

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-090953

(43)Date of publication of application : 04.04.1997

(51)Int.Cl.

G10H 1/36

G10H 1/28

(21)Application number : 07-241056

(71)Applicant : ROLAND CORP

(22)Date of filing : 20.09.1995

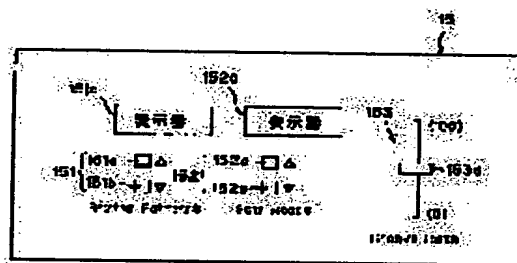
(72)Inventor : KISHIMOTO TETSUSHI

(54) ARPEGGIATER

(57)Abstract:

PROBLEM TO BE SOLVED: To easily play more musical arpeggios by generating musical performance information in timing according with time intervals stored in steps of a rhythm pattern table corresponding to its scanning.

SOLUTION: Plural kinds of rhythm pattern table are stored in a ROM and given numbers. Then a rhythm pattern number is set by operating an operator 151 for rhythm pattern setting and then a player selects an arbitrary rhythm pattern table. Further, plural kinds of scan mode tables are stored in the ROM and the player selects an arbitrary scan mode table by operating an operator 152 for scan mode setting. Then musical performance information is generated in timing according with the time intervals recorded in steps of the rhythm pattern table corresponding to its scanning.



LEGAL STATUS

[Date of request for examination] 23.05.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-90953

(43) 公開日 平成9年(1997)4月4日

(51) Int.Cl.⁸

識別記号

庁内整理番号

F I

技術表示箇所

G 1 0 H 1/36
1/28

G 1 0 H 1/36
1/28

審査請求 未請求 請求項の数16 O L (全 31 頁)

(21) 出願番号 特願平7-241056

(22) 出願日 平成7年(1995)9月20日

(71) 出願人 000116068

ローランド株式会社

大阪府大阪市北区堂島浜1丁目4番16号

(72) 発明者 岸本 哲史

大阪市北区堂島浜1丁目4番16号 ローラ
ンド株式会社内

(74) 代理人 弁理士 山田 正紀 (外2名)

(54) 【発明の名称】 アルペジエータ

(57) 【要約】

【課題】本発明は、押鍵情報を走査しその走査結果に従って複数の演奏情報を順次生成するアルペジエータに関し、より音楽的なアルペジオ演奏を簡単な操作で行なう。

【解決手段】各ステップ毎に、リズムの時間間隔、発音持続時間、発音の強さ係数が記憶されたリズムパターンテーブルを備え、そのリズムパターンテーブルに従ってアルペジオ演奏を行なう。

[Scan Mode Table]

SMODE_XXX_TBL[n]

| | |
|---------------|---|
| Scan Function | # |
| Scan Function | # |
| Scan Function | # |

:

| | |
|---------------|---|
| Scan Function | # |
|---------------|---|

SMODE_XXX_SIZE

| |
|-----------------|
| Scan Table Size |
|-----------------|

【特許請求の範囲】

【請求項 1】 複数の鍵それぞれに対応する記憶領域を有し、鍵が押鍵されたことを表わす押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込む押鍵情報記憶手段と、

リズムの 1 ステップ毎に、あるステップと、そのステップに続く次のステップとの間の時間間隔が、1 つのステップもしくは複数のステップそれぞれに記録されて成るリズムパターンテーブルを記憶するリズムパターンテーブル記憶手段と、

前記リズムパターンテーブルの各ステップを順次参照するとともに、該各ステップに対応して前記記憶領域を走査し、該走査により検出された押鍵情報に基づく楽音を表わす演奏情報を、前記リズムパターンテーブルの、該走査に対応するステップに記録された時間間隔に従ったタイミングで生成する演奏情報生成手段とを備えたことを特徴とするアルペジエータ。

【請求項 2】 複数の鍵それぞれに対応する記憶領域を有し、鍵が押鍵されたことを表わす押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込む押鍵情報記憶手段と、

リズムの 1 ステップ毎に、そのステップにおける楽音の特質を規定する情報が、1 つのステップもしくは複数のステップそれぞれに記録されて成るリズムパターンテーブルを記憶するリズムパターンテーブル記憶手段と、

前記リズムパターンテーブルの各ステップを順次参照するとともに、該各ステップに対応して前記記憶領域を走査し、該走査により検出された押鍵情報、および前記リズムパターンテーブルの、該走査に対応するステップに記録された、楽音の特質を規定する情報との双方に基づく楽音を表わす演奏情報を生成する演奏情報生成手段とを備えたことを特徴とするアルペジエータ。

【請求項 3】 前記楽音の特質を規定する情報が、楽音の発音持続時間を規定する情報であることを特徴とする請求項 2 記載のアルペジエータ。

【請求項 4】 前記楽音の特質を規定する情報が、楽音の発音の強さを規定する情報であることを特徴とする請求項 2 記載のアルペジエータ。

【請求項 5】 前記押鍵情報記憶手段が、鍵が押鍵されたことを表わすとともに押鍵の強さの情報を含む押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込むものであって、

前記演奏情報生成手段が、前記走査により検出された押鍵情報に含まれた押鍵の強さの情報に対応する発音の強さの情報を含む演奏情報を生成するものであることを特徴とする請求項 1 又は 2 記載のアルペジエータ。

【請求項 6】 前記押鍵情報記憶手段が、鍵が押鍵されたことを表わすとともに押鍵の強さの情報を含む押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込むものであって、

前記演奏情報生成手段が、前記走査により検出された押鍵情報に含まれた押鍵の強さの情報、および、前記リズムパターンテーブルの、該走査に対応するステップに記録された楽音の発音の強さを規定する情報の双方に基づいて定められた発音の強さの情報を含む演奏情報を生成するものであることを特徴とする請求項 4 記載のアルペジエータ。

【請求項 7】 前記リズムパターンテーブル記憶手段が、複数のリズムパターンテーブルを記憶するものであり、

前記演奏情報生成手段が、前記複数のリズムパターンテーブルの中から選択された 1 つのリズムパターンテーブルを参照するものであることを特徴とする請求項 1 又は 2 記載のアルペジエータ。

【請求項 8】 前記リズムパターンテーブル記憶手段が、前記複数のリズムパターンテーブルのうちの少なくとも一部のリズムパターンテーブルを書換え自在に、もしくは新たなリズムパターンテーブルを追加自在に記憶するものであることを特徴とする請求項 7 記載のアルペジエータ。

【請求項 9】 前記演奏情報生成手段が、楽音の特質の変更深さを規定する情報に基づいて変更された楽音を表わす演奏情報を生成するものであることを特徴とする請求項 1 又は 2 記載のアルペジエータ。

【請求項 10】 前記楽音の特質の変更深さを規定する情報であって、該変更深さが操作の度合に応じて定められてなる情報を生成する操作子を備えたことを特徴とする請求項 9 記載のアルペジエータ。

【請求項 11】 前記楽音の特質の変更深さを規定する情報が、楽音の発音持続時間の変更の度合を規定する情報であることを特徴とする請求項 9 記載のアルペジエータ。

【請求項 12】 前記楽音の特質の変更深さを規定する情報が、楽音の発音の強さの変更の度合を規定する情報であることを特徴とする請求項 9 記載のアルペジエータ。

【請求項 13】 前記押鍵情報記憶手段が、鍵が押鍵されたことを表わすとともに押鍵の強さの情報を含む押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込むものであって、

前記演奏情報生成手段が、前記走査により検出された押鍵情報に含まれた押鍵の強さの情報、前記リズムパターンテーブルの、該走査に対応するステップに記録された楽音の発音の強さを規定する情報、および楽音の発音の強さの変更の度合を規定する情報に基づいて定められた発音の強さの情報を含む演奏情報を生成するものであることを特徴とする請求項 4 記載のアルペジエータ。

【請求項 14】 複数の鍵それぞれに対応する記憶領域を有し、鍵が押鍵されたことを表わす押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込む押

鍵情報記憶手段と、

リズムの1ステップ毎に、あるステップと、そのステップに続く次のステップとの間の時間間隔が、1つのステップもしくは複数のステップそれぞれに記録されて成るリズムパターンテーブルを記憶するリズムパターンテーブル記憶手段と、

前記記憶領域を走査する方式を規定する複数種類のスキャン方式の中から選択された1つのスキャン方式、もしくは同一種類のスキャン方式の重複が許容された複数のスキャン方式それぞれが、1つのステップ、もしくは複数のステップそれぞれに記録されて成るスキャンモードテーブルを記憶するスキャンモードテーブル記憶手段と、

前記リズムパターンテーブルの各ステップおよび前記スキャンモードテーブルの各ステップを順次参照し、前記スキャンモードテーブルの各ステップに記録されたスキャン方式に従って前記記憶領域を走査し、該走査により検出された押鍵情報に基づく楽音を表わす演奏情報を、前記リズムパターンテーブルの、該走査に対応するステップに記録された時間間隔に従ったタイミングで生成する演奏情報生成手段とを備えたことを特徴とするアルペジエータ。

【請求項15】 複数の鍵それぞれに対応する記憶領域を有し、鍵が押鍵されたことを表わす押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込む押鍵情報記憶手段と、

リズムの1ステップ毎に、そのステップにおける楽音の特質を規定する情報が、1つのステップもしくは複数のステップそれぞれに記録されて成るリズムパターンテーブルを記憶するリズムパターンテーブル記憶手段と、

前記記憶領域を走査する方式を規定する複数種類のスキャン方式の中から選択された1つのスキャン方式、もしくは同一種類のスキャン方式の重複が許容された複数のスキャン方式それぞれが、1つのステップ、もしくは複数のステップそれぞれに記録されて成るスキャンモードテーブルを記憶するスキャンモードテーブル記憶手段と、

前記リズムパターンテーブルの各ステップおよび前記スキャンモードテーブルの各ステップを順次参照し、前記スキャンモードテーブルの各ステップに記録されたスキャン方式に従って前記記憶領域を走査し、該走査により検出された押鍵情報、および前記リズムパターンテーブルの、該走査に対応するステップに記録された、楽音の特質を規定する情報との双方に基づく楽音を表わす演奏情報を生成する演奏情報生成手段とを備えたことを特徴とするアルペジエータ。

【請求項16】 前記リズムパターンテーブル記憶手段および前記スキャンモードテーブル記憶手段が、それぞれ、複数のリズムパターンテーブルおよび複数のスキャンモードテーブルを記憶するものであり、

前記複数のリズムパターンテーブルのうちのいずれか1つのリズムパターンテーブルと前記複数のスキャンモードテーブルのうちのいずれか1つのスキャンモードテーブルとが対応づけられて成るスタイルを複数種類記憶するスタイル記憶手段を備え、

前記演奏情報生成手段が、所定のスタイル選択情報により選択されたスタイルにより対応づけられたリズムパターンテーブルおよびスキャンモードテーブルを参照するものであることを特徴とする請求項14又は15記載のアルペジエータ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、押鍵情報を走査しその走査結果に従って複数の演奏情報を順次生成するアルペジエータに関する。

【0002】

【従来の技術】従来の電子楽器には、いわゆるアルペジエータと呼ばれる機能が搭載されているものがある。このアルペジエータとは、時間の経過毎に、押鍵情報を、例えば音高の低い方から高い方へと順番に走査し、その結果に従って演奏情報を順次出力する機能である。

【0003】この機能によって、演奏者にとって本来演奏の困難なアルペジオ演奏を、単に複数の鍵を同時に押鍵するだけの簡単な演奏操作によって実現することができる。

【0004】

【発明が解決しようとする課題】しかしながら、従来のアルペジエータは、ある決まった一定の時間間隔で押鍵情報を走査してその一定の時間間隔で演奏情報を順次出力するものであり、このため発音タイミングは完全には均等になり、人間らしさがなく、音楽的に味気ないものになってしまう傾向があった。

【0005】また、楽音の発音持続時間や楽音の音量等の楽音の特質も均等であり、この点からも人間らしさがなく、音楽的に味気ないアルペジオ演奏となっていた。本発明は、従来のアルペジエータでは不可能であった、より音楽的なアルペジオ演奏を簡単な操作で行なうことのできるアルペジエータを提供することを目的とする。

【0006】

【課題を解決するための手段】上記目的を達成する本発明の第1のアルペジエータは、

(1-1) 複数の鍵それぞれに対応する記憶領域を有し、鍵が押鍵されたことを表わす押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込む押鍵情報記憶手段

(1-2) リズムの1ステップ毎に、あるステップと、そのステップに続く次のステップとの間の時間間隔が、1つのステップもしくは複数のステップそれぞれに記録されて成るリズムパターンテーブルを記憶するリズムパ

ターンテーブル記憶手段

(1-3) 上記リズムパターンテーブルの各ステップを順次参照するとともに、その各ステップに対応して上記記憶領域を走査し、その走査により検出された押鍵情報に基づく楽音を表わす演奏情報を、リズムパターンテーブルの、その走査に対応するステップに記録された時間間隔に従ったタイミングで生成する演奏情報生成手段を備えたことを特徴とする。

【0007】本発明の第1のアルペジエータによれば、リズムパターンテーブルの各ステップ毎にリズムの時間間隔が記録されており、その時間間隔に従ったタイミングで演奏情報が生成されるので、時間的に等分された単純なアルペジオ演奏だけでなく、さまざまなリズムのアルペジオ演奏が可能になる。また、上記目的を達成する本発明の第2のアルペジエータは、

(2-1) 複数の鍵それぞれに対応する記憶領域を有し、鍵が押鍵されたことを表わす押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込む押鍵情報記憶手段

(2-2) リズムの1ステップ毎に、そのステップにおける楽音の特質を規定する情報が、1つのステップもしくは複数のステップそれぞれに記録されて成るリズムパターンテーブルを記憶するリズムパターンテーブル記憶手段

(2-3) 上記リズムパターンテーブルの各ステップを順次参照するとともに、その各ステップに対応して上記記憶領域を走査し、その走査により検出された押鍵情報、および上記リズムパターンテーブルの、その走査に対応するステップに記録された情報との双方に基づく楽音を表わす演奏情報を生成する演奏情報生成手段を備えたことを特徴とする。

【0008】ここで、上記本発明の第2のアルペジエータにおいて、上記楽音の特質を規定する情報としては、例えば楽音の発音持続時間を規定する情報が採用される。この場合、スタッカートやテヌートの効果を、アルペジオ演奏の各楽音毎に独立して付加できるので、リズムの多彩なニュアンスやアーティキュレーションを付けることができる。

【0009】あるいは、上記本発明の第2のアルペジエータにおいて、上記楽音の特質を規定する情報として、楽音の発音の強さを規定する情報を採用してもよい。この場合、各楽音毎に強弱を付けることができ、さまざまなアクセントのアルペジオ演奏が実現する。上記本発明の第1のアルペジエータないし第2のアルペジエータにおいて、上記押鍵情報記憶手段が、鍵が押鍵されたことを表わすとともに押鍵の強さの情報を含む押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込むものであって、上記演奏情報生成手段が、前記走査により検出された押鍵情報に含まれた押鍵の強さの情報に対応する発音の強さの情報を含む演奏情報を生成するも

のであることが好ましい。

【0010】この場合、押鍵の強さにより示した演奏者の意図を楽音の発音の強さに反映させることができる。また、上記本発明の第2のアルペジエータにおいて、上記楽音の特質を規定する情報として、楽音の発音の強さを規定する情報を採用した場合に、上記押鍵情報記憶手段が、鍵が押鍵されたことを表わすとともに押鍵の強さの情報を含む押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込むものであって、上記演奏情報生成手段が、上記走査により検出された押鍵情報に含まれた押鍵の強さの情報、および、上記リズムパターンテーブルの、その走査に対応するステップに記録された楽音の発音の強さを規定する情報の双方に基づいて定められた発音の強さの情報を含む演奏情報を生成するものであることも好ましい態様である。

【0011】この場合、リズムパターンテーブルに記録された楽音の発音の強さを規定する情報を生かしつつ、演奏者の意図も楽音の発音の強さに反映させることができる。また、上記本発明の第1のアルペジエータないし第2のアルペジエータにおいて、上記リズムパターンテーブル記憶手段が、複数のリズムパターンテーブルを記憶するものであり、上記演奏情報生成手段が、複数のリズムパターンテーブルの中から選択された1つのリズムパターンテーブルを参照するものであることが好ましい。

【0012】このようにリズムパターンテーブルを複数記憶しておくことにより、演奏者の選択の幅が広がり、演奏者の好みやニュアンスに合ったアルペジオ演奏を行なうことができる。複数のリズムパターンテーブルを記憶する場合に、上記リズムパターンテーブル記憶手段が、それら複数のリズムパターンテーブルのうちの少なくとも一部のリズムパターンテーブルを書換え自在に、もしくは新たなリズムパターンテーブルを追加自在に記憶するものであることが好ましい。

【0013】少なくとも一部のリズムパターンテーブルを書換え自在もしくは追加自在としておくことにより、演奏者の好みに一層適合したアルペジオ演奏のリズムパターンを生成することができるとともに、書換え自在もしくは追加自在とすることに代えて演奏者の微妙な好みの相違それぞれに適合した多数のリズムパターンテーブルを用意しておく場合と比べ、メモリ容量の節約にもなる。

【0014】さらに、上記本発明の第1のアルペジエータないし第2のアルペジエータにおいて、上記演奏情報生成手段が、楽音の特質の変更深さを規定する情報に基づいて変更された楽音を表わす演奏情報を生成するものであることも好ましい態様である。上記の楽音の特質の変更深さを規定する情報としては、例えば、楽音の発音持続時間の変更の程度を規定する情報や、あるいは楽音の発音の強さの変更の程度を規定する情報が採用され

る。

【0015】この場合、リズムパターンテーブル全体を書き換えたり、異なるリズムパターンのテーブルを追加したりすることなく、アルペジオ演奏におけるスタッカートやテヌートの効果、あるいはアクセントの効果などの微妙な調整が可能となる。上記の楽音の特質の変更深さを規定する情報は外部から入力されるものであってもよいが、楽音の特質の変更深さを規定する情報であっても、その変更深さが操作の度合に応じて定められてなる情報を生成する操作子を備えると、その操作子を操作することで、上記の効果をリアルタイムで変更することができ、より音楽的なアルペジオ演奏が可能となる。

【0016】また、上記本発明の第2のアルペジエータにおいて、リズムパターンテーブルに記録された楽音の特質を規定する情報として楽音の発音の強さを規定する情報を採用した場合に、上記押鍵情報記憶手段が、鍵が押鍵されたことを表わすとともに押鍵の強さの情報を含む押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込むものであって、上記演奏情報生成手段が、上記走査により検出された押鍵情報に含まれた押鍵の強さの情報、上記リズムパターンテーブルの、その走査に対応するステップに記録された楽音の発音の強さを規定する情報、および楽音の発音の強さの変更の度合を規定する情報の三者に基づいて定められた発音の強さの情報を含む演奏情報を生成するものであることも好ましい態様である。

【0017】この場合、リズムパターンテーブルに記録された楽音の発音の強さを規定する情報を生かしつつ、演奏者の意図も楽音の発音の強さに反映させることができ、しかも、リズムパターンテーブル全体を書き換えたり、異なるリズムパターンのテーブルを追加したりすることなく、アルペジオ演奏におけるアクセントの効果の微妙な調整が可能となる。

【0018】また、本発明の第3のアルペジエータは、
(3-1) 複数の鍵それぞれに対応する記憶領域を有し、鍵が押鍵されたことを表わす押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込む押鍵情報記憶手段

(3-2) リズムの1ステップ毎に、あるステップと、そのステップに続く次のステップとの間の時間間隔が、1つのステップもしくは複数のステップそれぞれに記録されて成るリズムパターンテーブルを記憶するリズムパターンテーブル記憶手段

(3-3) 上記記憶領域を走査する方式を規定する複数種類のスキャン方式の中から選択された1つのスキャン方式、もしくは同一種類のスキャン方式の重複が許容された複数のスキャン方式それぞれが、1つのステップ、もしくは複数のステップそれぞれに記録されて成るスキャンモードテーブルを記憶するスキャンモードテーブル記憶手段

(3-4) 上記リズムパターンテーブルの各ステップおよび上記スキャンモードテーブルの各ステップを順次参照し、上記スキャンモードテーブルの各ステップに記録されたスキャン方式に従って上記記憶領域を走査し、そのにより検出された押鍵情報に基づく楽音を表わす演奏情報を、リズムパターンテーブルの、該走査に対応するステップに記録された時間間隔に従ったタイミングで生成する演奏情報生成手段を備えたことを特徴とする。

【0019】さらに、本発明の第4のアルペジエータは、

(4-1) 複数の鍵それぞれに対応する記憶領域を有し、鍵が押鍵されたことを表わす押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込む押鍵情報記憶手段

(4-2) リズムの1ステップ毎に、そのステップにおける楽音の特質を規定する情報が、1つのステップもしくは複数のステップそれぞれに記録されて成るリズムパターンテーブルを記憶するリズムパターンテーブル記憶手段

(4-3) 上記記憶領域を走査する方式を規定する複数種類のスキャン方式の中から選択された1つのスキャン方式、もしくは同一種類のスキャン方式の重複が許容された複数のスキャン方式それぞれが、1つのステップ、もしくは複数のステップそれぞれに記録されて成るスキャンモードテーブルを記憶するスキャンモードテーブル記憶手段

(4-4) 上記リズムパターンテーブルの各ステップおよび上記スキャンモードテーブルの各ステップを順次参照し、スキャンモードテーブルの各ステップに記録されたスキャン方式に従って上記記憶領域を走査し、その走査により検出された押鍵情報、およびリズムパターンテーブルの、その走査に対応するステップに記録された、楽音の特質を規定する情報の双方に基づく楽音を表わす演奏情報を生成する演奏情報生成手段を備えたことを特徴とする。

【0020】本発明の上記第3のアルペジエータおよび第4のアルペジエータは、それぞれ、本発明の第1のアルペジエータおよび第2のアルペジエータの構成に加え、さらに、上記スキャンモードテーブル記憶手段を備え、複数のスキャン方式、例えば上昇スキャン方式、下降スキャン方式等の中から選択された1つもしくは複数の、スキャン方式をスキャンモードテーブルに記録してにおいて、実際のアルペジオ演奏にあたってはそのスキャンモードテーブルの各ステップに記録されたスキャン方式を順次参照しながら、参照したスキャン方式に応じて押鍵情報を走査して演奏情報を生成するものであるため、上述した、それぞれ、本発明の第1のアルペジエータおよび第2のアルペジエータの利点に加え、さらに、従来と同様の複数の鍵を同時に押える等の簡単な操作で、スキャンモードテーブルに記録されたスキャン方式

のパターンに応じ極めて複雑な、あるいは音楽的な意味のあるアルペジオ演奏が実現する。

【0021】ここで、上記本発明の第3のアルペジエータないし第4のアルペジエータにおいて、上記リズムパターンテーブル記憶手段および上記スキャンモードテーブル記憶手段が、それぞれ、複数のリズムパターンテーブルおよび複数のスキャンモードテーブルを記憶するものであり、このアルペジエータが、上記複数のリズムパターンテーブルのうちのいずれか1つのリズムパターンテーブルと上記複数のスキャンモードテーブルのうちのいずれか1つのスキャンモードテーブルとが対応づけられて成るスタイルを複数種類記憶するスタイル記憶手段を備え、上記演奏情報生成手段が、所定のスタイル選択情報により選択されたスタイルにより対応づけられたリズムパターンテーブルおよびスキャンモードテーブルを参照するものであることが好ましい。

【0022】この場合、演奏者は、リズムパターンとスキャンモードとの組合せをいちいち意識することなく、スタイルを選ぶことで自分の好みのリズムパターンとスキャンモードが選択され、使い勝手の良いアルペジエータが実現する。

【0023】

【発明の実施の形態】以下、本発明の実施形態について説明する。図1は、本発明のアルペジエータの一実施形態の回路構成図である。ここには、CPU10、アドレスバス11およびデータバス12でそのCPU10と接続された、RAM13、ROM14、パネル15、およびCPU10にクロックの発生を通知するクロックタイマ16、MIDI規格に準拠したデータを入力するデータ入力端子(MIDI IN)17、MIDI規格に準拠したデータを出力するデータ出力端子(MIDI OUT)18が備えられている。

【0024】ROM14は、読出し専用のメモリエリアであり、ここには、CPU10で実行されるプログラムや、そのプログラムで参照される各種テーブルが格納されている。プログラムやテーブルの詳細については後述する。RAM13は、CPU10で実行されるプログラムのワーキングエリアとして使用される。ここでは、このRAM13は電池でバックアップされており、電源を切ってもそのときのメモリ内容は保存される。メモリの内容について後述する。

【0025】パネル15は、操作スイッチと表示器を持ち、パラメータを独立して選択しそのパラメータに値を設定することができる。設定された値はRAM13内のメモリに記憶される。図2は、パネル15を示した図である。このパネルでは、リズムパターン番号(Rhythm Pattern#)、スキャンモード番号(Scan Mode#)、およびグルーブレート(Groove Rate)を設定することができる。

【0026】リズムパターンは、本実施形態では、発音

タイミング、デュレーション、ベロシティ係数のパターンを組み合わせたものである。リズムパターン設定用操作子151は、値をインクリメントする操作ボタン151aと値をデクリメントする操作ボタン151bとのペアから成り、それらの操作ボタン151a、151bで設定した値は表示器151cに表示されるとともに、上述したように、RAM13に格納される。

【0027】スキャンモードは押鍵情報をどのように走査するかというスキャン方式のセットをいう。スキャンモード設定用操作子152は、上記リズムパターン設定用操作子151と同様に、値をインクリメントする操作ボタン152aと値をデクリメントする操作ボタン151bとのペアから成り、それらの操作ボタン151a、151bで設定した値は、表示器152cに表示されるとともに、RAM13に格納される。

【0028】グルーブレートは、いわゆる「ノリ」の程度を定めるものであり、グルーブレート設定用操作子153はスライダから成り、そのスライドボタン153aを一番下に動かすとグルーブレートパラメータに値「0」が設定され、一番上に動かすとグルーブレートパラメータに値「100」が設定され、それらの中間に設定すると設定された位置に応じた、0~100の間の値が設定される。この設定された値は、RAM13に格納される。

【0029】各パラメータの詳細については後述する。図1に戻って説明を続行する。クロックタイマ16は、一定の周期でクロックを発生し、クロック発生のために、CPUに対し、クロックの発生を通知する。本実施形態では、96クロック分の時間が4分音符の発音時間と定められており、このクロック周期によってアルペジオのテンポが決定される。

【0030】データ入力端子(MIDI IN)17には、例えば図示のように鍵盤20が接続され、鍵盤20の演奏により生成された押鍵(ノートオン)情報、離鍵(ノートオフ)情報が入力される。また、その鍵盤20には、ホールドペダル21が接続されており、そのホールドペダル21の操作により生成されたホールドオン情報、ホールドオフ情報(これらを合わせてホールド情報と称する。)も、MIDIデータの種類として、データ入力端子17を経由して入力される。

【0031】CPU10は、データ入力端子17から入力される押鍵情報、離鍵情報を受信してアルペジオ演奏用の演奏情報を順次生成し、その演奏情報をデータ出力端子18から外部に向けて出力する。データ出力端子18には、例えば図示のように音源30が接続されており、その音源30では受信した演奏情報に基づいて楽音信号が生成される。その楽音信号は、例えば音源30に内蔵された、あるいは音源30に接続されたアンプ、スピーカ等から成るサウンドシステムにより、楽音として空間に放音される。

【0032】以下、先ず後述するプログラムで参照される。ROM内に記憶された各種テーブルについて説明し、次いで、RAM内に確保された各種メモリについて説明し、その後、それらのテーブルやメモリを用いたプログラムについて説明する。図3は、リズムパターンテーブルの基本パターンを示す図である。リズムパターンテーブルは、その各ステップがリズムの各ステップに対応しており、各ステップには、そのステップとそのステップに続く次のステップとの間の時間間隔（ステップタイム；Step Time）、そのステップのリズム音の持続時間（デュレーション；Duration）、そのステップのリズム音の発音の強さを規定するベロシティ係数（Velo. Coef.）が記録されており、最後にそのリズムパターンテーブルのサイズ（ステップ数；Step Size）が記録されている。1ステップ分のメモリ容量は、ステップタイムが2バイト、デュレーションが2バイト、ベロシティ係数が1バイトの合計5バイトであり、ステップ数は、1バイトである。ステップタイム、デュレーションの欄は、前述したクロック（4分音符＝96クロック）を単位とした数値が記録されている。

【0033】このリズムパターンテーブルはROM内に複数種類記録されており、各リズムパターンテーブルには番号が付されている。図2に示すリズムパターン設定用操作子151を操作してそのリズムパターン番号を設定することにより、演奏者が任意のリズムパターンテーブルを選択することができる。ここでは、リズムパターンテーブル、およびそのステップ数は、それぞれ、RHY_XXX_TBL[n]、RHY_XXX_SIZEと表記される。XXXはそのリズムパターンテーブルの名称を表わし、[n]のnは、そのリズムパターンテーブル内部のステップ番号（ $0 \leq n < \text{RHY_XXX_SIZE}$ ）である。

【0034】またリズムパターンテーブル名XXXの第nステップに記録されたステップタイム、デュレーション、ベロシティ係数は、それぞれ、
 RHY_XXX_TBL[n]. stepTime
 RHY_XXX_TBL[n]. duration
 RHY_XXX_TBL[n]. veloCoef
 と表記される。

【0035】次にリズムパターンテーブルの各例について説明する。図4は、4分音符用のリズムパターンテーブルおよびそのステップ数を示した図である。このリズムパターンテーブルは4分音符用であることを表わす「4」という名称を有しており、RHY_4_TBLと表記される。RHY_4_TBL[0]. stepTime、RHY_4_TBL[0]. duration、RHY_4_TBL[0]. veloCoefは、それぞれ96（4分音符のクロック数）、92、127である。ステップ数はRHY_4_SIZEと表記される。

このリズムパターンテーブルRHY_4_TBLは1ステップのみで成り立っているため、ステップ数RHY_4_SIZEは「1」である。

【0036】このリズムパターンテーブルRHY_4_TBL、および以下に示すリズムパターンテーブルRHY_XXX_TBLのそれぞれには、それぞれ異なるリズムパターン番号が付されており、この図4に示すリズムパターンテーブルRHY_XXX_TBLのリズムパターン番号は「0」である。図5、図6、図7、図8、図9は、それぞれ、16分音符用（名称：16）、ワルツ（WALTZ）用、シャッフル（SHUFFLE）用、ディミニッシュ（DIM）用、レゲエ（REGGAE）用のリズムパターンテーブルおよびそのステップ数を示した図である。リズムパターン番号はこの順に、1、2、3、4、5が付されている。これらの構造については上述の説明から明らかであるため、詳細説明は省略する。

【0037】図10は、スキャンモードテーブルの基本パターンを示す図である。スキャンモードテーブルの各ステップには、押鍵情報を走査する方式（スキャン方式）の番号（スキャンファンクション番号；Scan Function#）が記録されている。このスキャンモードテーブル1ステップ分のメモリ容量は1バイトであり、ステップ数も1バイトである。

【0038】このスキャンモードテーブルは、ROM14に複数種類記憶されており、図2に示すスキャンモード設定用操作子152を操作することにより、演奏者が任意のスキャンモードテーブルを選択することができる。ここで、スキャンモードテーブル、およびそのステップ数は、それぞれSMODE_XXX_TBL[n]、SMODE_XXX_SIZEと表記される。XXXはそのスキャンモードテーブルの名称を表わし、[n]のnは、そのスキャンモードテーブル内部のステップ番号（ $0 \leq n < \text{SMODE_XXX_SIZE}$ ）である。

【0039】本実施形態における各スキャン方式（Scan Function）は以下のとおりである。

（1）上昇スキャン方式

押鍵情報を、音高の低い方から高い方へと走査する。ここでは、これを、SCAN_Uと表記する。

【0040】（2）下降スキャン方式

音高の高い方から低い方へと走査する。ここではこれを、SCAN_Dと表記する。

（3）上昇下降スキャン方式

音高の低い方から高い方へと走査し、最高音に達したら、今度は音高の高い方から低い方へと走査する。ここでは、これをSCAN_UDと表記する。

【0041】（4）ランダムスキャン方式

ランダムに走査する。ここではこれを、SCAN_Rと表記する。

(5) 押鍵順スキャン方式

押鍵情報の入力順に走査する。ここではこれを、SCAN_Oと表記する。

(6) 和音スキャン方式

現在押鍵されているものすべてを走査する。ここではこれを、SCAN_Cと表記する。

【0042】(7) 最低音スキャン方式

現在押鍵されているもののうちノートナンバのもっとも小さなもの(最低音)だけを走査する。ここではこれを、SCAN_Bと表記する。

(8) 最高音スキャン方式

現在押鍵されているもののうちノートナンバのもっとも大きなもの(最高音)だけを走査する。ここではこれを、SCAN_Tと表記する。

【0043】(9) 最低音抜き上昇スキャン方式

最低音を除いて、音高の低い方から高い方へと走査する。ここでは、これをSCAN_U_WO_B (Scam up without bass)と表記する。

(10) 最低音抜きランダムスキャン方式

最低音を除いて、ランダムに走査する。ここでは、これをSCAN_R_WO_Bと表記する。

【0044】(11) 最低音抜き和音スキャン方式

最低音を除いて、現在押鍵されているものをすべて走査する。ここでは、これをSCAN_C_WO_Bと表記する。

(12) 最高音抜き上昇スキャン方式

最高音を除いて、音高の低い方から高い方へと走査する。ここでは、これをSCAN_U_WO_Tと表記する。

【0045】次にスキャンモードテーブルの各例を示す。図11は、上昇スキャン用のスキャンモードテーブル、およびそのステップ数を示した図である。このスキャンモードテーブルは、上昇スキャン用であることを表わす'UP'という名称を表わしており、SMODE_UP_TBLと表記される。ステップ数は、SMODE_UP_SIZEと表記される。このスキャンモードテーブルSMODE_UP_TBLは、上昇スキャン方式SCAN_Uが記録された1ステップのみで成り立っているため、ステップ数SMODE_UP_SIZEは'1'である。このスキャンモードテーブルSMODE_UP_TBL、および以下に示す各スキャンモードテーブルSMODE_XXX_TBLのそれぞれには、それぞれ異なるスキャンモード番号が付されており、この図11に示すスキャンモードテーブルSMODE_UP_TBLのスキャンモード番号は'0'である。

【0046】図12、図13、図14、図15、図16、図17、図18、図19は、それぞれ下降スキャン用(DOWN)、上昇下降スキャン用(UP_DW)、ランダムスキャン用(RANDOM)、和音スキャン用(CHORD)、押鍵順スキャン用(ORDER)、ワ

ルツ用(WALTZ)、レゲエ用(REGGAE)、三味線用(SHAMI)のスキャンモードテーブルおよびそのステップを示した図である。

【0047】スキャンモード番号は、この順に、1、2、3、4、5、6、7、8が付されている。これらの構造については、上述の説明から明らかであるため、詳細説明は省略する。次に各種パラメータ値等が格納されるRAM内のメモリについて説明する。図20はリズムパターンパラメータを示す図である。このリズムパターンパラメータは1バイトのメモリであり、このメモリには、図2に示すパネル15のリズムパターン設定用操作子151の操作により設定されたリズムパターン番号が格納される。ここではこれを、PRM_RHYTHMと表記する。

【0048】図21は、スキャンモードパラメータを示す図である。このスキャンモードパラメータも、1バイトのメモリであり、このメモリには、図2に示すパネル15のスキャンモード設定用操作子152の操作により設定されたスキャンモード番号が格納される。ここでは、これを、PRM_SMODEと表記する。図22は、グルーブレートパラメータを示す図である。このグルーブレートパラメータも1バイトのメモリであり、このメモリには図2に示すパネル15のグルーブレート設定用操作子の操作により設定されたグルーブレートの値が格納される。この値の範囲は、前述したように、0~100である。ここではこれを、PRM_GROOVEと表記する。

【0049】図23は、ノートバッファを示す図である。このノートバッファは、128バイトの配列であり、各1バイトは、鍵盤20(図1参照)の各鍵(ノートナンバ)に対応しており、押鍵情報を受信すると、その押鍵情報中の押鍵強さを表わす情報(ノートオンベロシティ)が、その押鍵情報中のノートナンバに対応する領域に格納される。対応する鍵が離鍵(ノートオフ)されている場合、あるいはそれまで押鍵されていて離鍵された場合は、その領域には'-1'が格納される。

【0050】ここでは、このノートバッファの各領域をNOTEBUF[n]と表記する。nはノートナンバを表わす。図24は、プレイバッファを示す図である。このプレイバッファは、図23に示すノートバッファと同様128バイトの配列であり、各1バイトは、各ノートナンバに対応している。ただし、このプレイバッファには、図23に示すノートバッファとは異なり、ホールド情報を加味したノートオンベロシティが記憶される。すなわち、図1に示す鍵盤20に接続されたホールドペダル21が踏まれると、データ入力端子17を経由してホールドオン情報が入力され、ホールドペダル21が離されるとデータ入力端子17を経由してホールドオフ情報が入力されるが、ホールドオン情報を受信すると、ホールドオフ情報を受信する迄の間、この図24に示すプレ

イバッファに格納された押鍵情報の消去が禁止される。ホールドオフ情報が入力されると、プレイバッファにはそのときのノートバッファの内容がコピーされ、以後、次にホールドオン情報が入力されない限り、プレイバッファは常にノートバッファと同一の内容を維持する。アルペジオ演奏を行なうにあたっては、ホールド情報が加味されたベロシティが配列された、このプレイバッファが走査される。ここでは、このプレイバッファの領域を、PLAYBUF[n]と表記する。nはノートナンバを表わす。

【0051】図25は、カレントプレイノートナンババッファを示す図である。このカレントプレイノートナンババッファは1バイトのメモリであり、このメモリには、アルペジオ演奏中の現在のステップのノートナンバが格納される。ここではこれを、CUR_NOTEと表記する。図26は、ホールドバッファを示す図である。このホールドバッファは1バイトのメモリであり、ここには0~127の数値のホールド情報が格納される。図1に示すホールドペダル21を操作すると、データ入力端子17からは、その操作量に応じた0~127の数値で表わされるホールド情報が入力され、そのホールド情報がこのホールドバッファに格納される。ここでは、この数値が64以上であればホールドオン情報、64未満であればホールドオフ情報とみなされる。ここではこれを、HOLDと表記する。

【0052】図27は、オーダバッファを示す図である。このオーダバッファは、1ステップが2バイトからなる16ステップの配列であり、各ステップには、ノートナンバとそのノートナンバのベロシティが格納される。このオーダバッファには、押鍵された鍵のノートナンバとその押鍵時のベロシティが押鍵順に格納され、離鍵されたときは、その鍵のノートナンバおよびベロシティが抹消され先頭詰めに並べ直される。ここではこれを、ORDER[n]と表記する。nは、ステップ番号(n=0~15)である。特に、第nステップのノートナンバのみ、ベロシティのみを指すときは、それぞれORDER[n]:note、ORDER[n]:velと表記される。

【0053】図28は、オーダライトカウンタを示す図である。このオーダライトカウンタは1バイトのメモリであり、このメモリには、押鍵情報を次に受信した場合に、その受信した押鍵情報を、図27に示すオーダバッファORDER[]の何番目のステップに格納するかを示す値が格納される。このオーダライトカウンタのとり得る値は0~16である。0~15は、オーダバッファORDER[]の各ステップに対応し、16は、オーダバッファORDER[]が満杯であることを意味する。ここではこれを、ORDER_WRと表記する。

【0054】図29は、オーダポジションカウンタを示す図である。このオーダポジションカウンタは1バイト

のメモリであり、このメモリには、アルペジオ演奏中の現在のステップの楽音が図27に示すオーダバッファORDER[]の何番目のステップに対応する楽音であることを示す値が格納される。このオーダポジションカウンタのとり得る値は-1~15である。0~15は、図27に示すオーダバッファORDER[]の各ステップに対応し、-1はアルペジオ演奏の一連のステップをこれから開始するタイミング(前回の一連のステップが終了したタイミング)であることを意味している。ここではこれを、CUR_ORDERと表記する。

【0055】図30は、リズムパターンテーブルポジションカウンタを示す図である。このリズムパターンテーブルポジションカウンタは1バイトのメモリであり、このメモリは、現在選択されているリズムパターンテーブル(図3、および図4~図9の各例を参照)中のステップを指し示すポインタである。このポインタはアルペジオ演奏中の現在のリズムステップに対応している。このリズムパターンテーブルポジションカウンタのとり得る値は0以上、かつ現在選択されているリズムパターンテーブルのステップ数RHY_XXX_SIZE未満である。ここではこれを、CUR_RHYと表記する。

【0056】図31は、スキャンモードテーブルポジションカウンタを示す図である。このスキャンモードテーブルポジションカウンタは1バイトのメモリであり、現在選択されているスキャンモードテーブル(図10、および図11~図19の各例を参照)中のステップを指し示すポインタである。このポインタも、アルペジオ演奏中の現在のリズムステップに対応している。スキャンモードテーブルポジションカウンタのとり得る値は、0以上、かつ現在選択されているスキャンモードテーブルのステップ数SMODE_XXX_SIZE未満である。ここでは、これをCUR_SMODEと表記する。

【0057】図32は、クロックカウンタを示す図である。このクロックカウンタは2バイト(16ビット)のメモリであり、図1に示すクロックタイマ16がクロックを1つ発生するたびにこのクロックカウンタの値が'1'ずつ増加する。このクロックカウンタの値の範囲は、0000h~FFFFhであり、FFFFhの次は0000hに戻り、電源がオンである間、無限に巡回する。このクロックの値の増加に基づいてアルペジオが進行する。ここではこれを、CLOCKと表記する。

【0058】図33は、ネクストクロックバッファを示す図である。このネクストクロックバッファは2バイト(16ビット)のメモリであり、このメモリには、アルペジオ演奏の次のステップの時刻が格納される。すなわち、図32に示すCLOCKが増加して、このネクストクロックバッファに格納されている値に達すると、押鍵情報を走査してノートオンデータを送信する。1回のステップの処理が終了すると、このネクストクロックバッファの値は次のステップの時刻に更新される。このネク

ストクロックバッファのとり得る値の範囲は0000h~FFFFhである。ここではこれを、NEXTCLKと表記する。

【0059】図34は、ノートオフリザベーションバッファを示す図である。このノートオフリザベーションバッファは3バイト×16の配列であり、このノートオフリザベーションバッファには、ノートオフすべきノートナンバと、それをノートオフデータとして実際に送信すべきタイミングとがペアで格納される。この送信タイミングは図32のCLOCKの値で表わされる。CLOCKが順次増加していったこのノートオフリザベーションバッファに格納されている送信タイミングのうちのいずれかの送信タイミングに到達すると、その送信タイミングとペアで格納されているノートナンバについてノートオフデータが送信される。このノートオフリザベーションバッファ中の有効なデータが格納されていない領域、あるいはノートオフデータを送信し終わった領域には「-1」が格納される。

【0060】ここではこれを、OFFRSV[n]と表記する。nは、ノートナンバとその送信タイミングとのペアのn番目の格納領域を意味する。特にOFFRSV[n]の中ノートナンバ、送信タイミングを指し示すときは、それぞれ、OFFRSV[n].note, OFFRSV[n].clockと表記される。図35は、エンドフラグを示す図である。このエンドフラグは、ここでは1バイトのメモリで構成されているが、フラグの性質上1ビットで構成してもよい。このエンドフラグは「0」と「1」との2値をとり、「0」は、アルペジオ演奏中の1回のステップの発音のための走査が未了であることを示し、「1」は、1回のステップの発音のための走査が終了したことを示している。ここではこれを、END_FLGと表記する。

【0061】図36は、ベースノートバッファを示す図である。このベースノートバッファは1バイトのメモリであり、このメモリには、ホールド情報を加味した押鍵中の鍵のうちの最低音の鍵のノートナンバが格納される。ここではこれを、LOと表記する。図37は、トップノートバッファを示す図である。このトップノートバッファは1バイトのメモリであり、このメモリにはホールド情報を加味した押鍵中の鍵のうちの最高音の鍵のノートナンバが格納される。ここではこれを、HIと表記する。

【0062】以上説明した各メモリの他にも作業領域として使用するメモリもあるが、後述するプログラムの説明で足りるため、ここでは、ここに説明した以外のメモリについて1つずつ取りあげた説明は省略する。以下に、これまで説明したテーブルやメモリを用いてCPUで実行されるプログラムについて説明する。

【0063】図38は、電源オン時に動作を開始し、電源オフ時まで動作し続けるゼネラルプログラムのフロー

チャートである。電源が投入されると、先ず所定の初期化が行なわれ(ステップ38_1)、それ以後、クロックタイマ16(図1参照)からクロックの発生が通知されたか否か(ステップ38_2)、パネル15(図1、図2参照)が操作されいずれかのパラメータがエディットされたか否か(ステップ38_4)、ノートオンデータ(押鍵情報)が入力されたか否か(ステップ38_6)、ノートオフデータ(離鍵情報)が入力されたか否か(ステップ38_8)、ホールドデータ(ホールド情報)が入力されたか否か(ステップ38_10)が順次循環的にモニタされ、各ステップ38_2、38_4、38_6、38_8、38_10で各イベントが発生したことが認識されると、それぞれ、クロック処理(ステップ38_3)、エディット処理(ステップ38_5)、ノートオン処理(ステップ38_7)、ノートオフ処理(ステップ38_9)、ホールドデータ処理(ステップ38_11、38_12、38_13)が行なわれる。

【0064】ホールドデータ処理では、先ずHOLD(図26参照)にホールド値が格納され(ステップ38_11)、そのホールド値が、ホールドオンを表わす64以上であるか否かが判定され(ステップ38_12)、ホールドオフを表わす63以下の場合、ホールドオフ処理が行なわれる(ステップ38_13)。図39は、初期化処理ルーチンのフローチャートである。この初期化処理ルーチンは、図38に示すゼネラルプログラムのステップ38_1で実行される。

【0065】この初期化処理では、先ず、押鍵時のペロシティが押鍵順に格納されるオーダバッファORDER[] (図27参照)がクリアされ、そのオーダバッファORDER[]の格納ポインタORDER_WR(図28参照)が「0」(ORDER[]の先頭)に初期化され、さらに、オーダポジションカウンタCUR_ORDER(図29)が、アルペジオ演奏の一連のステップをこれから開始するタイミングであることを表わす「-1」に初期化される(ステップ39_1)。

【0066】次いで、クロックタイマ16(図1参照)で発生したクロックをカウントするクロックカウンタCLOCK(図32参照)が「0」に初期化され(ステップ39_2)、次の演奏情報の生成のタイミングが格納されるネクストクロックバッファNEXTCLK(図33参照)も「0」に初期化される(ステップ39_3)。さらに、ノートオフすべきノートナンバやそのノートオフすべきタイミングが格納されるノートオフリザベーションバッファOFFRSV[] (図34参照)がクリアされ(ステップ39_4)、鍵盤20(図1参照)の鍵の配列に対応した配列を有し押鍵された鍵のペロシティが格納されるノートバッファNOTEBUF[] (図23参照)がクリアされ(ステップ39_5)、さらに、ホールド情報を加味したペロシティが格

納されるプレイバッファPLAYBUF[] (図24参照) がクリアされる (ステップ39_6)。さらにホールド値が格納されるホールドバッファHOLD (図26参照) が '0' にクリアされ (ステップ39_7)、さらに、スキャナをリセットするリセットスキャナルーチンが実行される (ステップ39_8)。

【0067】図40は、リセットスキャナルーチンのフローチャートである。リセットスキャナルーチンは、図39に示す初期化ルーチンのステップ39_8で実行される。ここでは、現在アルペジオ演奏中のノートナンバが格納されるカレントプレイノートナンババッファCUR_NOTE (図25参照) に空きを表わす '−1' が格納され、オーダバッファORDER[] (図27参照) の、現在アルペジオ演奏中のステップを指し示すオーダポジションカウンタCUR_ORDER (図29参照) にも、空きを表わす '−1' が格納され、さらに、スキャンモードテーブル、リズムパターンテーブルの各ポインタであるスキャンモードテーブルポジションカウンタCUR_SMODE (図31参照)、リズムパターンテーブルポジションカウンタCUR_RHY (図30参照) が、先頭を指し示す '0' に初期化される。さらに、クロックCLOCK (図32参照) に1を加えた値がネクストクロックNEXTCLK (図33参照) に格納される。CLOCKに '1' を加えた値をNEXTCLKとする理由は、後述するようにNEXTCLK=CLOCKのときに発音される (ノートオンデータが送信される) が、このプログラムが動作中にCLOCKがインクリメントされてCLOCKが発音開始タイミングを示すNEXTCLKを越えてしまい、その発音が行なわれないことが生じる可能性をなくするためである。

【0068】図40に示すリセットスキャナルーチンでは、さらに、演奏中のアルペジオ演奏の1回ステップの発音のための操作が未了か終了かを示すエンドフラグEND_FLGに、未了を示す '0' が格納される。図41は、クロック処理ルーチンのフローチャートである。このクロック処理ルーチンは、図38に示すゼネラルプログラムのステップ38_3で実行される。

【0069】このクロック処理ルーチンでは、まずCLOCK (図32参照) がインクリメントされる (ステップ41_1)。次に、OFFRSV[] (図34参照) をサーチし、そこに現在のタイミングで送信すべきノートオフデータが存在するかどうかを調べ、現在のタイミングで送信すべきノートオフデータが存在する場合にそのノートオフデータを送信する (ステップ41_2~41_8)。

【0070】具体的には、まずiに '0' を置いて (ステップ41_2)、OFFRSV[i].noteに有効なノートオフデータが格納されているか (0~127) 否か (−1) を調べ (ステップ41_3)、'−1' の場合はiをインクリメントして (ステップ41_

7)、iが16に達するまで (ステップ41_8)、そのサーチが行なわれる。ステップ41_3で有効なノートオフデータの格納が確認されるとステップ41_4に進み、OFFRSV[i].clockが現在のCLOCKと等しいかどうか判定され、等しい場合にそのノートオフデータを演奏情報として送信し (ステップ41_5)、そのノートオフデータが格納されていた領域のOFFRSV[i].noteに '−1' を書き込む (ステップ41_6)。

【0071】ノートオフデータのサーチが終了すると、今度は送信すべきノートオンデータを見つけに行く処理を行なう (ステップ41_9~41_17)。具体的には、まず現在のCLOCKがNEXTCLKに達したか否か、すなわちノートオンデータを送信すべきタイミングに達したか否かが判定され (ステップ41_9)、未だそのタイミングに達していないときはそのまま終了する。

【0072】NEXTCLK=CLOCKのときは、ステップ41_10に進み、押鍵情報を走査するスキャンノートオンルーチンが実行される (ステップ41_10)。このスキャンノートオンルーチンの詳細は後述するが、このスキャンノートオンルーチンでは、送信すべきノートナンバのうちの1つがNTに格納される。NT=−1は、送信すべきノートが存在しないことを意味している。ステップ41_11においてNT=−1ではない、すなわち送信すべきノートが存在すると判定されると、ステップ41_12に進み、ノートオフ予約ルーチンが実行される。このノートオフ予約ルーチンの詳細についても後述するが、このノートオフ予約ルーチンでは、これから送信しようとするノートオンデータに対応するノートオフデータをOFFRSV[] (図34参照) に格納することにより、ノートオフ予約を行なう。OFFRSV[] が満杯のときは、NTに、ノートオフ予約ができないことを示す '−1' が格納される。

【0073】ステップ41_13ではNT=−1かどうかを調べることにによりノートオフ予約が行なわれたか否かを知り、ノートオフ予約不能であったときはノートオンデータを送信せずにステップ41_10に戻り、送信すべき次のノートオンデータのサーチを行なう。ステップ41_13でNT≠−1であったとき、すなわち正常にノートオフ予約が行なわれたときは、ステップ41_14に進み、ベロシティ生成ルーチンが実行される。このベロシティ生成ルーチンの詳細についても後述する。このベロシティ生成ルーチンでは、後述する演算により送信しようとするノートオンデータのベロシティが生成される。その後ステップ41_15においてそのノートオンデータが送信され、ステップ41_10に戻り、現在のタイミングでさらに送信すべきノートオンデータが存在するかどうかのサーチが行なわれる。

【0074】ステップ41_10でNTに '−1' が格

納された場合、すなわち現在のタイミングで送信すべきノートオンデータが存在しない（もしくは、現在のタイミングで送信すべきノートオンデータは全て送信してしまった）場合、ステップ41_11を経由してステップ41_16に進む。ステップ41_16では、ネクストクロック更新ルーチンが実行される。このネクストクロック更新ルーチンの詳細は後述するが、ここでは、ノートオンデータ送信の次のタイミングがNEXTCLKに格納される。

【0075】その後、ステップ41_17に進み、スキャナ更新ルーチンが実行される。このスキャナ更新ルーチンでスキャナの更新が行なわれる。このスキャナ更新ルーチンの詳細についても後述する。図42は、スキャンノートオンルーチンのフローチャートである。このスキャンノートオンルーチンは、図41のクロック処理ルーチンのステップ41_10で実行される。

【0076】このスキャンノートオンルーチンでは、先ずスキャンモード番号が格納されているPRM_SMODE（図21参照）が参照され（ステップ42_1）、そのPRM_SMODEに格納されているスキャンモード番号に応じたスキャンモードテーブル（図10、および図11～図19の各例参照）中の現在のステップ番号CUR_SMODEに格納されたスキャン方式（スキャンファンクション番号）SMODE_XXX_TBL [CUR_MODE] が作業領域funcに格納される（ステップ42_2）。次いで、そのfuncに格納されたスキャンファンクション番号がSCAN_U, SCAN_D, SCAN_UD, ……、SCAN_U_WO_Tのいずれを指しているかに応じて、それぞれ、図示の、スキャンアップルーチン（ステップ42_4）、スキャンダウンルーチン（ステップ42_5）、スキャンアップダウンルーチン（ステップ42_6）、……、スキャンアップウィズアウトトップルーチン（42_15）が実行される。

【0077】以下では、代表的なものとして、スキャンアップルーチン（ステップ42_4）、スキャンオーダールーチン（ステップ42_8）、スキャンコードルーチン（ステップ42_9）、スキャンベースルーチン（ステップ42_10）、スキャンコードウィズアウトベース（ステップ42_14）について説明する。図43は、スキャンアップルーチンのフローチャートである。このスキャンアップルーチンは、前述した上昇スキャン方式を実現するルーチンである。

【0078】ここでは、先ず、END_FLG（図35参照）が‘1’か‘0’か、すなわち、今回の1ステップ分の発音のための走査が既に終了しているか否かが判定され（ステップ43_1）、END_FLG=1の場合、すなわち、今回の走査が既に終了している場合は、ステップ43_2に進み、NTに‘-1’が格納されて終了する。ただし、今回のステップで最初にこのスキャ

ナアップルーチンが実行される際はEND_FLG=0である。

【0079】ステップ43_1でEND_FLGが‘1’でなかったときは、ステップ43_3に進み、PLAYBUF[]（図24参照）に1つでもノートオンデータが存在するか否かが判定され、ノートオンデータが1つも存在しないときは、発音しようがないので、ステップ43_2に進み、NTに‘-1’が格納されて終了する。

【0080】PLAYBUF[]にノートオン情報が存在することが認識されると、次に、上昇スキャン方式に従って、次に送信すべきノートオンデータが走査される（ステップ43_4～43_7）。具体的には、CUR_NOTE（図25参照）は、直前に送信したノートオンデータのノートナンバが格納されているため、そのCUR_NOTEを次のノートナンバに進め（ステップ43_4）、そのノートナンバが128に達すると（ステップ43_5）、CUR_NOTEに‘0’が格納される（ステップ43_6）、このようにして更新されたCUR_NOTEを用いて、PLAYBUF [CUR_NOTE] に有効なベロシティデータが格納されているか否かが判定される（ステップ43_7）。PLAYBUF [CUR_NOTE] = -1、すなわちそこには有効なベロシティデータが格納されていないときは、ステップ43_4に進み、CUR_NOTEが再度インクリメントされ、PLAYBUF[]の有効なベロシティが格納されている領域を見い出すべく、音高の低い方から高い方へと走査される。

【0081】このようにして、有効なベロシティデータが格納されているPLAYBUF[]が見い出されると、そのときのCUR_NOTEがNTに格納され（ステップ43_8）、PLAYBUF [NT] に格納されている、CUR_NOTEのベロシティデータがVLに格納され（ステップ43_9）、上昇スキャン方式では1つのステップでは1つのノートオンデータしか送信しないためEND_FLGに走査の終了を示す‘1’が格納される（ステップ43_10）。

【0082】図44は、スキャンオーダールーチンのフローチャートである。このスキャンオーダールーチンは、前述した押鍵順スキャン方式を実現するルーチンである。ステップ44_1、44_2は、図43に示すスキャンアップルーチンのステップ43_1、43_2と同様であり、説明は省略する。ステップ44_3では、ORDER_WR=0か否かが調べられる。ORDER_WRは、図28と参照して説明したように、押鍵情報を押鍵順に格納するORDER[]（図27参照）のポインタであり、ORDER_WR=0ということは、ORDER[]が空であって、押鍵順アルペジオ演奏を行なうことができないことを意味しており、この場合もステップ44_2に進み、NTに‘-1’が格納されて終了す

る。

【0083】ステップ44_3でORDER_WR=0ではない、すなわちORDER [] が空ではないと判定されると、ステップ44_4に進んでCUR_ORDER (図29参照) がインクリメントされ、CUR_ORDERがORDER [] の書き込まれる最大ステップを越えたか否かが判定されて (ステップ44_5)、越えた場合は、CUR_ORDERに0が格納される (ステップ44_6)。このようにして値が進められたCUR_ORDERを用いて、次に発音すべきノートナンバーORDER [CUR_ORDER]. noteがNTに格納され (ステップ44_7)、そのノートナンバーのベロシティORDER [CUR_ORDER]. veloがVLに格納される (ステップ44_8)。この押鍵順スキャン方式においても、1つのステップでは1つのノートオンデータしか送信しないため、END_FLGに、走査の終了を示す '1' が格納される (ステップ44_9)。

【0084】図45は、スキャンコードルーチンのプログラムである。このスキャンコードルーチンは、前述した和音スキャン方式を実現するルーチンである。ステップ45_1、45_2、45_3は、図43に示すスキャンアップルーチンのステップ43_1、43_2、43_3と同様であり、説明は省略する。今回のステップにおいて、このスキャンコードルーチンが第1回目と呼ばれたときは、CUR_NOTEには '-1' が格納されており、したがってステップ45_4、45_5、45_6のループでは、CUR_NOTE=0から始まり、CUR_NOTE=127まで走査される。その走査の途中で、PLAYBUF [CUR_NOTE] に '-1' 以外の値、すなわち有効なベロシティデータが格納されていることを見出すと、ステップ45_7に進み、NTにそのノート番号CUR_NOTEを格納し、VLにそのノートナンバーのベロシティデータPLAYBUF [NT] を格納して (ステップ45_8)、このルーチンを一旦抜ける。

【0085】和音スキャン方式では、PLAYBUF [] の全域を走査し有効なベロシティデータの格納された全てのノートオンデータを送信する必要があるため、CUR_NOTE \geq 128に達する前にこのスキャンコードルーチンを抜けるときはEND_FLGには '1' には格納しない。このスキャンコードルーチンを抜けると、図42に示すスキャンノートオンルーチンも抜け、図41のクロック処理ルーチンのステップ41_10を抜けることになる。NT=-1ではないときは、ステップ41_12、41_13、41_14、41_15と進んで1つのノートオンデータの送信が行なわれ、再度ステップ41_10に戻り、図42のスキャンノートオンルーチンを経由して、図45のスキャンコードルーチンが再度実行される。そのときには、再度実行

を開始したときの、すなわち前回このルーチンを抜けたときCUR_NOTEの次の値CUR_NOTEから走査が行なわれる (ステップ45_4)。ステップ45_5においてCUR_NOTE \geq 128、すなわちPLAYBUF [] の走査が終了したことが判定されると、ステップ45_9に進んでNTに '-1' が格納され、さらにステップ45_10に進んで、END_FLGに '1' が格納される。

【0086】図46は、スキャンベースルーチンのプログラムである。このスキャンベースルーチンは、前述した最低音スキャン方式を実現するルーチンである。ステップ46_1、46_2、46_3は、図43に示すスキャンアップルーチンのステップ43_1、43_2、43_3と同様であり、説明は省略する。ステップ46_4では、最低音LO (図36参照) がNTに格納される。LOへの最低音の格納処理については後述する。

【0087】ステップ46_5では、その最低音LOのベロシティPLAYBUF [NT] がVLに格納される。さらにステップ46_6において、END_FLGに '1' が格納される。最低音は1つしか存在しないからである。図47は、スキャンコードウィズアウトベースルーチンのプログラムである。このルーチンは、前述した最低音抜き和音スキャン方式を実現するルーチンである。

【0088】ステップ47_1、47_2、47_3は、図43に示すスキャンアップルーチンのステップ43_1、43_2、43_3と同様であり、説明は省略する。ステップ47_4では、PLAYBUF [] にノートオンの数が1つだけ存在するか、それとも複数存在するかが判定される、ノートオンの数が1つだけの場合、本来は最低音抜き和音スキャン方式は成立しないが、ここでは、その唯一のノートオンのノートナンバーを送信することとし、ステップ47_5に進んで、NTにそのノートのノートナンバーを格納し、そのノートナンバーのベロシティPLAYBUF [NT] をVLに格納し (ステップ47_6)、END_FLGに '1' を格納する (ステップ47_7)。

【0089】ステップ47_4においてノートオンの数が複数存在することが確認されるとステップ47_8に進む。今回のステップにおいて、このスキャンコードウィズアウトベースルーチンが第1回目と呼ばれたときは、図45に示すスキャンコードルーチンの場合と同様に、CUR_NOTEには '-1' が格納されている。CUR_NOTEに '-1' を格納する処理については後述する。したがって、ここでは、PLAYBUF [] の全域にわたって、音高の低い方から高い方へと順次走査される。ここでの走査の手順 (ステップ47_8~47_15) は、図45に示すスキャンコードルーチンにおける走査の手順 (ステップ45_4~45_10) と同様であるが、ステップ47_11において、CUR_

NOTEが最低量LOであるか否かを判定し、最低音の場合にそのノートオンを無視するようにしている点のみが異なる。詳細説明は省略する。尚、LOに最低音を格納する処理については後述する。

【0090】以上では、図42に示すスキャンノートオンルーチン中で実行される、スキャンアップルーチン（ステップ42_4）、スキャンオーダールーチン（ステップ42_8）、スキャンコードルーチン（ステップ42_9）、スキャンベースルーチン（42_10）、スキャンコードウィズアウトベースルーチン（ステップ42_13）について説明した。スキャンダウンルーチン（ステップ42_5）、スキャンアップダウンルーチン（ステップ42_6）、スキャンランダムルーチン（ステップ42_7）、スキャントップルーチン（ステップ42_11）、スキャンアップウィズアウトベースルーチン（ステップ42_12）、スキャンランダムウィズアウトベースルーチン（ステップ42_13）、スキャンアップウィズアウトトップルーチン（42_14）は、前述した、それぞれ、下降スキャン方式、上昇下降スキャン方式、ランダムスキャン方式、最高音スキャン方式、最低音抜き上昇スキャン方式、最低音抜きランダムスキャン方式、最高音抜き上昇スキャン方式を実現するルーチンであるが、上述した各種のルーチン（図43～図47）から自明であるため、ここではそれらについての図示および説明は省略する。

【0091】図48は、ノートオフ予約ルーチンのプログラムである。このノートオフ予約ルーチンは、図41に示すクロック処理ルーチンのステップ41_12で実行される。このノートオフ予約ルーチンでは、ノートオフ予約に用いられるOFFRSV[]（図34参照）をサーチしてその空いている領域を見つけ、一方、リズムパターンテーブルから読み出したデュレーションと、ステップタイムと、グルーブレート設定用操作子153（図2参照）の操作により設定されたグルーブレートPRM_GROOVEとに従ってデュレーションを計算し、そのデュレーションに基づいてノートオフのタイミングを求めて、そのノートオフのタイミングをOFFRSV[]の空き領域に設定することにより、ノートオフ予約を行なう。

【0092】すなわち、先ずステップ48_1においてiに'0'を設定し、ステップ48_2においてOFFRSV[i].note=-1であるか否か、すなわち、OFFRSV[i]が空いているか否かが判定される。OFFRSV[i].note=-1ではないとき、すなわちOFFRSV[i]が空いていないときは、iがインクリメントされ（ステップ48_3）、iがOFFRSV[]の配列の大きさを越えたか否かが判定され（ステップ48_4）、その配列以内であるときはステップ48_2に戻ってインクリメントされた新たなiに対してOFFRSV[i].note=-1か否

かに判定される。これを繰り返し、空き領域を発見できないままi=16に達すると、ステップ48_5に進み、NTに、OFFRSV[]が満杯であることを示す'-1'を格納して終了する。

【0093】ステップ48_2においてOFFRSV[i].note=-1、すなわち空き領域を見つけたときは、その空き領域OFFRSV[i].noteに、図41のクロック処理ルーチンのステップ41_10で得られたノートナンバNTが格納される（ステップ48_6）。次いで、リズムパターン番号PRM_RHYTHM（図20参照）に応じて（ステップ48_7）、そのリズムパターン番号PRM_RHYTHMに対応するリズムパターンテーブルRHY_XXX_TBL[]（図3、および図4～図9の各例を参照）の現在のステップ番号CUR_RHY（図30参照）のステップに格納されている、デュレーションRHY_XXX_TBL[CUR_RHY].durationおよびステップタイムRHY_XXX_TBL[CUR_RHY].stepTimeが読み出されて、それぞれD1、D2に格納される（ステップ48_8）。

【0094】ステップ48_9では、デュレーションD1、ステップタイムD2、およびグルーブレートPRM_GROOVE（図22参照）に基づいて、そのノートナンバのデュレーションDURATIONが、
$$DURATION = (D1 - D2) \times PRM_GROOVE / 100 + D2$$
の計算により求められる。

【0095】この計算結果は、PRM_GROOVE=0のときは、DURATION=D2、すなわちステップタイムと同値となり、そのアルペジオ演奏にテヌートのような効果を与え、PRM_GROOVE=100のときは、DURATION=D1、すなわちリズムパターンテーブルRHY_XXX_TBL[]に格納されたデュレーションのままとなる。このリズムパターンテーブルRHY_XXX_TBL[]にスタッカートのような短いデュレーションを格納しておく、図2に示すグルーブレート設定用操作子153の操作により、スタッカートのようなアルペジオ演奏からテヌートのようなアルペジオ演奏まで、各音のデュレーションを連続的に変えることができる。このようにして求めたDURATIONは、CLOCK（図32参照）と加算されて、OFFRSV[i].clockに格納される（ステップ48_10）。

【0096】図49は、ベロシティ生成ルーチンのフローチャートで、図50は、図49に示すベロシティ生成ルーチンで生成されるベロシティの説明図である。図49に示すベロシティ生成ルーチンは、図41に示すクロック処理ルーチンのステップ（41_14）で実行される。図49に示すベロシティ生成ルーチンでは、リズムパターン番号PRM_RHYTHM（図20参照）が参

照され(ステップ49_1)、そのリズムパターン番号PRM_RHYTHMに対応するリズムパターンテーブルRHY_XXX_TBL(図3、および図4~図9の各例参照)の現在のステップCUR_RHY(図38参照)のベロシティ係数RHY_XXX_TBL[CUR_RHY]・velocityが読み出されてCOEFに格納される(ステップ49_2)。その後、ステップ49_3において、ベロシティ係数COEFと、押鍵により生成されたベロシティVLとの間で、

$$VL2 = VL \times COEF / 127$$

の計算が行なわれ、さらにステップ49_4において、 $VL - (VL - VL2) \times PRM_GROOVE / 100$

の計算が行なわれて、その計算の結果求められたベロシティが再度VLに格納される。

【0097】ここでは、図50に示すように、図2に示すグループレート設定用操作子153の操作により設定されたグループレートPRM_GROOVEがPRM_GROOVE=0の時、リズムパターンテーブルRHY_XXX_TBL[]に格納されたベロシティにかかわらず、押鍵ベロシティがそのまま採用され、PRM_GROOVE=100のときは、(押鍵ベロシティ×ベロシティ係数)が採用され、グループレート設定用操作子153をその中間に設定すると、その位置に応じて、押鍵ベロシティと、(押鍵ベロシティ×ベロシティ係数)との間で連続的に変化するベロシティが採用される。

【0098】このようにグループレートを変えることにより、いわゆる「ノリ」の程度を変化させることができる。図41のクロック処理ルーチンのステップ41_14では、上記のようにしてベロシティが求められ、ステップ41_15では、このようにして求められたベロシティデータを伴ったノートオンデータが送信される。

【0099】図51は、ネクストクロック更新ルーチンのフローチャートである。このネクストクロック更新ルーチンは、図40に示すクロック処理ルーチンのステップ41_16で実行される。このネクストクロック更新ルーチンでは、まずリズムパターン番号PRM_RHYTHM(図20参照)が参照され、そこに格納されたリズムパターン番号PRM_RHYTHMに応じたリズムパターンテーブルRHY_XXX_TBL[]の現在のステップ番号CUR_RHY(図30参照)のステップに格納されているステップタイムRHY_XXX_TBL[CUR_RHY]・stepTimeが読み出されてSTEPに格納され(ステップ51_2)、CLOCK(図32参照)にそのSTEPが加算されてNEXTCLK(図33参照)に格納される(ステップ51_3)。これにより、次の発音タイミングがNEXTCLKに設定される。

【0100】図52、53は、それぞれ、スキヤナ更新ルーチンのフローチャートの前半部分、後半部分であ

る。このスキヤナ更新ルーチンは、図41に示すクロック処理ルーチンのステップ41_17で実行される。このスキヤナ更新ルーチンでは、まず、前回のノートオンデータの走査の最後でEND_FLGに「1」が格納されたのを解除してEND_FLGに「0」を格納し(ステップ52_1)、次に、リズムパターンテーブルの現在のステップCUR_RHY(図30参照)が更新される(ステップ52_2~52_6)。具体的には、ステップ52_2においてCUR_RHYがインクリメントされ、次いで、リズムパターン番号PRM_RHYTHM(図20参照)が参照されそのリズムパターン番号PRM_RHYTHMに応じたリズムパターンテーブルのステップ数RHY_XXX_SIZEが読み出されてSIZEに格納され(ステップ52_4)、CUR_RHY ≤ SIZEであるか否か、すなわち、CUR_RHYがインクリメントされた(ステップ52_2)結果、現在使用中のリズムパターンテーブルRHY_XXX_TBL[]のステップ数を越えたか否かが判定され、越えていないときはそのままステップ52_7に進み、越えたときは、CUR_RHYに「0」が格納されることにより先頭のステップに戻った上で、ステップ52_7に進む。

【0101】ステップ52_7~52_11では、スキャンモードテーブルの現在のステップCUR_SMODE(図31参照)が更新される。すなわち、まずステップ52_7でCUR_SMODEがインクリメントされ、その後スキャンモード番号PRM_SMODE(図21参照)が参照され(ステップ52_8)、そのスキャンモード番号PRM_SMODEに対応するスキャンモードテーブルのステップ数SMODE_XXX_SIZEが読み出されてSIZEに格納され(ステップ52_9)、CUR_SMODE ≥ SIZEの場合に(ステップ52_10)、CUR_SMODEが先頭(ステップ0)に戻される(ステップ52_11)。

【0102】次に図53に示すステップ53_1に進み、スキャンモード番号PRM_SMODE(図21参照)が参照され、そのスキャンモード番号PRM_SMODEに対応したスキャンモードテーブルSMODE_XXX_TBL[]の、図52のステップ52_7~52_11で更新されたステップCUR_SMODEのスキャン方式(スキャンファンクション番号)SMODE_XXX_TBL[CUR_SMODE]が読み出されて、funcに格納される。ステップ53_3、53_4では、それぞれ、funcが和音スキャン方式SCAN_Cであるか否か、および最低音抜き和音スキャン方式SCAN_C_WO_Bであるか否かが判定され、SCAN_CもしくはSCAN_C_WO_Bの場合は、CUR_NOTE(図25参照)に「-1」が格納される。SCAN_CもしくはSCAN_C_WO_Bの場合、CUR_NOTEに「-1」を格納しておくことに

よって、ノートオンデータの走査のときに音高の低い方から高い方に向かってPLAYBUF [] の領域が走査される(図45、図47参照)。図54は、エディットルーチンのフローチャートである。このエディットルーチンは、パネル(図2参照)が操作された際に、図38に示すゼネラルプログラムのステップ38_5で実行される。

【0103】ステップ54_1では、リズムパターンが変更されたか否かが判定される。リズムパターンの変更は、図2に示すリズムパターン設定用操作子151を操作することにより行われる。リズムパターン設定用操作子151を操作することによりリズムパターンが変更される。ステップ54_2において、PRM_RHYTHM(図20参照)に、変更後の新たなリズムパターン番号が格納され、ステップ52_3に進み、リセットスキヤナルーチン(図40参照)が実行されてスキヤナのリセットが行われる。

【0104】ステップ54_4では、スキャンモードが変更されたか否かが判定される。スキャンモードの変更は、図2に示すスキャンモード設定用操作子152を操作することにより行われる。スキャンモード設定用操作子152が操作されることによりスキャンモードが変更されると、ステップ54_5において、PRM_SMD(図21参照)に変更後の新たなスキャンモード番号が格納され、ステップ52_3でスキヤナのリセットが行われる。

【0105】ステップ54_6では、グルーブレートが変更されたか否かが判定される。グルーブレートの変更は、図2に示すグルーブレート設定用操作子153の操作により行われる。グルーブレート設定用操作子153の操作によりグルーブレートが変更されると、ステップ54_7に進み、PRM_GROOVE(図22参照)にグルーブレートの新たな値が格納される。

【0106】図55は、ノートオン処理ルーチンのフローチャートである。このノートオン処理分は、ノートオンデータを受信した際に、図38に示すゼネラルブロックプログラムのステップ38_7で実行される。ステップ55_1では、受信したノートオンデータのうちのノートナンバがNTに格納され、ベロシティがVLに格納される。このベロシティVLは、NOTEBUF[NT](図23参照)およびPLAYBUF[NT](図24参照)に格納される(ステップ55_2、55_3)。次に、ORDER_WR(図28参照)が16未満端か否かが判定され、16未満のときは、押鍵順バッファORDER[] (図27参照)に空きがあるため、ORDER[ORDER_WR]:noteにノートナンバNTが格納され、ORDER[ORDER_WR]:veloにベロシティVLが格納される。ステップ55_7では、今回入力されたノートオンデータが全て離鍵(ノートオフ)されていた状態から最初に受信したノ

ートオンデータであるか否かが判定される。すなわち、具体的には、PLAYBUF[] に格納されている有効なベロシティデータは、ステップ55_3で今回格納したベロシティ1個のみであるか否かが検出される。最初のノートオンデータであった場合はステップ55_8に進み、スキヤナのリセットが実行される。その後、PLAYBUF[] が走査されて、最低音LO、最高音HIが更新される(ステップ55_9、55_10)。

【0107】図56は、ノートオフ処理ルーチンのフローチャートである。このノートオフ処理ルーチンは、ノートオフデータを受信した際に、図38に示すゼネラルプログラムのステップ38_9で実行される。ステップ56_1では、受信したノートオフデータのノートナンバがNTに格納され、NOTEBUF[NT](図23参照)に、有効なデータが格納されていないことを表す'-1'が格納される(ステップ56_2)。ステップ56_3では、HOLD(図26参照)が64未満(ホールドオフ)か否かが判定され、ホールドオン(64以上)のときはそのまま終了する。ホールドオフ(64未満)のときは、PLAYBUF[NT](図24参照)に'-1'が格納され(ステップ56_4)、押鍵順バッファORDER[] (図27参照)から今回のノートオフデータに対応するノートオンデータを削除するデリートオーダールーチンが実行され(ステップ56_5)、さらにデリートオーダールーチンにより削除が行われた押鍵順バッファORDER[] を前詰めにするバックオーダールーチンが実行される(ステップ56_6)。ステップ56_7、56_8ではPLAYBUF[] がサーチされて最低音LO、最高音HIが更新される。

【0108】図57は、デリートオーダールーチンのフローチャートである。このデリートオーダールーチンは、図56に示すノートオフ処理ルーチンのステップ56_5で実行される。ここではORDER[] (図27参照)がサーチされ、ノートオフデータのノートナンバNTの存在がサーチされ(ステップ57_1~57_5)、ORDER[i]=NTの場合、ORDER[i]:noteおよびORDER[i]:veloの双方に'-1'が格納される。複数のORDER[i]にNTが格納されていた場合、その全てについて'-1'が格納される。

【0109】図58は、バックオーダールーチンのフローチャートである。このバックオーダールーチンは、図56に示すノートオフ処理ルーチンのステップ56_6で実行される。ここでは、押鍵順バッファORDER[] がサーチされ、'-1'が格納されている領域をなくすように前詰めされる。具体的には、ステップ58_1において、i、j双方に初期値'0'が格納され、ステップ58_2においてORDER[i]:noteに'-1'が格納されているか否かが調べられる。ORDER[i]:noteに有効なデータ(すなわち'-1'以外)が格納されていたときは、ステップ58_3に進

み、ORDER [i]. note、ORDER [i]. veloが、それぞれ、ORDER [j]. note、ORDER [j]. veloに転記され、ステップ58__4でjがインクリメントされる。ORDER [] の前詰めに伴って、押鍵順アルペジオ演奏における現在のステップCUR_ORDER (図29参照) も変更する必要があることから、ステップ58__5ではCUR_ORDERがiか否か調べられ、CUR_ORDER=iのときは、CUR_ORDERにj-1が格納される (ステップ58__6)。ステップ58__7ではiがインクリメントされ、ステップ58__8でiがORDER_WR (図28参照) 以下か否か判定され、 $i \leq \text{ORDER_WR}$ の場合ステップ58__2に戻って、更新されたiについてのORDER [i]. noteが'-1'か否か調べられる。

【0110】ORDER [] のポインタORDER_WR (図28参照) もORDER [] の前詰めに伴って変更する必要があり、ステップ58__9では、ORDER_WRにjが格納される。図59は、ホールドオフ処理ルーチンのフローチャートである。このホールドオフ処理ルーチンはホールドデータを受信した際に、図38に示すゼネラルプログラムのステップ38__13で実行される。このホールドオフ処理ルーチンが実行されるときは、図38のステップ38__12の判定により、HOLDが64未満 (ホールドオフ) であることがわかっている。

【0111】図59に示すホールドオフ処理ルーチンが実行されると、押鍵、離鍵をそのまま反映したNOTE BUF [] (図23参照) が、その全域にわたって、ホールド情報を加味した押鍵情報を格納するPLAYBUF [] (図24参照) に転記される (ステップ59__1~59__4)。次いで、押鍵順バッファORDER [] (図27参照) の再構築を行うリメイクオーダールーチンが実行される (ステップ59__5)。ORDER [] が再構築された後、ステップ59__6、59__7では、PLAYBUF [] がサーチされて最低音LO、最高音HIの更新が行われる。

【0112】図60は、リメイクオーダールーチンのフローチャートである。このリメイクオーダールーチンは図59に示すホールドオフ処理ルーチンのステップ59__5で実行される。ここでは、PLAYBUF [] を音高の低い方から高い方と順にサーチするために、NTに、先ず初期値'0'が格納される (ステップ60__1)。

【0113】次いで、PLAYBUF [NT] が読み出されてVLに格納され (ステップ60__2)、そのVLが'-1'か否か、すなわちPLAYBUF [NT] に有効なベロシティデータが格納されていた (-1以外) か否か (-1) が判定される。VL=-1のときは、ステップ60__4に進み、デリートオーダールーチン (図57参照) が実行され、このデリートオーダールーチンの中

で、ORDER [] にノートナンバNTのノートオンデータが格納されていた場合にそれを削除する (具体的には'-1'を格納する)。ステップ60__5ではNTがインクリメントされ、ステップ60__6では、NTが128未満か否か判定され、NTが128未満のときはステップ60__2に戻って新たなNTについてPLAYBUF [NT] が読み出される。

【0114】このようにして、PLAYBUF [] の全域がサーチされ、ORDER [] から、VL=-1が格納されているPLAYBUF [NT] に対応するノートナンバNTが全て削除される。その後、ステップ60__7において、バックオーダールーチン (図29参照) が実行されて、ORDER [] の'-1'が格納された領域をなくすようにORDER [] が前詰めされる。これにより、ORDER [] は、現在押鍵中の鍵のみについて、押鍵順に押鍵情報が並んだ状態となる。

【0115】本実施形態におけるアルペジエータは、以上説明したように構成されており、図2に示すリズムパターン設定用操作子151およびスキャンモード設定操作子152により設定された、リズムパターンテーブル (図3、および図4~図9の各例参照) およびスキャンモードテーブル (図10および図11~図19の各例参照) を用いて、複数の鍵を同時に押鍵し、あるいは順次に押鍵して押鍵したままとし、あるいはホールドペダルを踏んで押鍵を記憶させておくという簡単な動作で、音楽的に意味の深い、種々のアルペジオ演奏を行うことができる。また、図2に示すグループレート操作子151を操作することにより、'ノリ'の程度を簡単に変更することができる。

【0116】次に、本発明のアルペジエータの他の実施形態について説明する。ただし、これ以降では、上述した実施形態と共通的な部分については図示および説明は省略し、上述の実施形態との相違点のみ、図示および説明を行う。図61は、図2のパネルに付加されるベロシティポリウムを示した図、図62は、図2に示すパネルに図61に示したベロシティポリウムが付加された構成において、図49に示すベロシティ生成ルーチンに代えて採用されるベロシティ生成ルーチンのフローチャートである。

【0117】この実施形態では、図61に示すようなベロシティポリウム154がパネル15 (図2参照) に付加される。このベロシティポリウム154は、手でつまんで回転させることができ、反時計まわりにいっぱい回転させると'0'、時計まわりにいっぱい回転させると'127'、それらの中間位置ではその位置に応じた値が出力され、RAM13 (図1参照) 中のVELOCITY_VOLUMEに格納される。このVELOCITY_VOLUMEは、図62のフローチャートで説明するように、アルペジオ演奏におけるベロシティデータの生成にかかわっている。

【0118】図62に示すベロシティ生成ルーチンでは、図49のベロシティ生成ルーチンと同様に、リズムパターン番号PRM_RHYTHMが参照され(ステップ62_1)、そのリズムパターン番号PRM_RHYTHMに対応するリズムパターンテーブルPHY_XXX_TBL(図3、および図4~図9の各例参照)の現在のステップCUR_RHY(図38参照)のベロシティ係数RHY_XXX_TBL[CUR_RHY]・velocityが読み出されて、COEFに格納される(ステップ62_2)。

【0119】その後、図49に示すベロシティ生成ルーチンと異なり、VELOCITY_VOLUME=0か否かが判定される(ステップ62_3)。VELOCITY_VOLUME=0の場合は、図49のベロシティ生成ルーチンと同様であり、ステップ62_5に進んで、ベロシティ係数COEFと、押鍵により生成されたベロシティVLとの間で、

$$VL2 = VL \times COEF / 127$$

の計算が行われ、さらにステップ62_6において、 $VL - (VL - VL2) \times PRM_GROOVE / 100$

の計算が行われて、その計算の結果求められたベロシティが再度VLに格納される。

【0120】一方、ステップ62_3において、VELOCITY_VOLUMEが'0'ではないと判定されると、ステップ62_4に進み、そのVELOCITY_VOLUMEがVLに格納される。すなわち、この場合は、押鍵により生成されたベロシティは無視され、その押鍵により生成されたベロシティに代わり、図61に示すベロシティボリューム154により設定されたVELOCITY_VOLUMEが採用される。

【0121】このように、押鍵ベロシティの代わりに固定のベロシティを用いるようにすると、押鍵強さに関係なく、一定の強さを持つアルペジオ演奏が実現できる。すなわち、演奏状態に左右されず安定した再現性のあるアルペジオ演奏を行うことができる。図63は、図2のパネルに付加されるスタイルスイッチを示した図、図64は、図2に示すパネルに図63に示したスタイルスイッチが付加された構成において、図54に示すエディットルーチンに代えて採用されるエディットルーチンのフローチャートである。

【0122】図63に示すスタイルスイッチ155は、ジャンル毎に複数(この例では3個)のボタン155a、155b、155cから成り、各ボタン155a、155b、155cを押すと、それぞれワルツ(WALTZ)、レゲエ(REGGAE)、三味線(SHAMISEN)のイメージに合致した、リズムパターン番号とスキャンモード番号とのセットが選択されるようになっている。

【0123】図64に示すエディットルーチンでは、先

ず、ステップ64_1において、スタイルスイッチ155(図63参照)のいずれかのボタン155a、155b、155cが押されることによりスタイルが変更されたか否かが判定される。スタイルが変更されたことが認識されると、その変更後のスタイルに応じて(ステップ64_2)、ワルツの場合はPRM_RHYTHM(図20参照)、PRM_SMODE(図21参照)にそれぞれ'2'(図6参照)、『6』(図17参照)が格納され、レゲエの場合は、PRM_RHYTHM、PRM_SMODEにそれぞれ'5'(図9参照)、『7』

(図18参照)が格納され、三味線の場合はPRM_RHYTHM、PRM_SMODEにそれぞれ'1'(図5参照)、『8』(図19参照)が格納される。図64のエディットルーチンの他のステップ64_4~64_10は、図54に示すエディットルーチンのステップ54_1~54_7と同様であるため、説明は省略する。

【0124】このように、本実施形態では、例えばワルツのボタン154aを押したときには、ワルツに最もふさわしいと一般的に考えられるリズムパターンとスキャンモードが同時に選択されて設定される。したがって演奏者は、リズムパターンとスキャンモードとの組合せをいちいち考えることなく、自分の選びたいスタイルをワンタッチで選ぶことができる。

【0125】なお、上記実施形態ではアルペジオ演奏に使用する音色に関しては特に触れなかったが、選択されたスタイルに対応してそのスタイルの演奏に適した音色を自動的に選択するようにしてもよい。例えば、ワルツ、レゲエのスタイルが選択された場合にはワルツ、レゲエの楽曲を演奏する際によく用いられる楽器の音色をそれぞれ選択し、三味線のスタイルが選択された場合には三味線の音色を選択するようにしてもよい。尚、上記各実施形態では、ホールド情報はデータ入力端子17

(図1参照)を経由して入力されるが、パネル15にホールドオン情報とホールドオフ情報とを切換え入力するスイッチを設けてもよく、あるいはアルペジオ自体にホールド情報生成用ペダル、ないしそのペダル接続用のジャックを備えてもよい。また、上記各実施形態では、鍵盤および音源が外部に接続されているが、鍵盤もしくは音源のうちの一方もしくは双方を内蔵し、あるいは一体的に構成してもよく、それに加え、さらに外部と送受信できるようにしてもよい。

【0126】また、上記各実施形態では、グループプレート設定用操作子153を1個のみ備え、その1個のグループプレート設定用操作子153で設定したグループプレートPRM_GROOVEによってアルペジオ演奏の各ステップのデュレーションとベロシティとの双方を変更しているが、それらの双方を変更できる構成になっている必要はなく、それらのうちの一方のみ変更できても、音楽的に十分意味のある変更が可能である。あるいは、より微妙な調整を演奏者に委ねるために、その1個のグル

ープレート設定用操作子153を、デュレーション変更用とベロシティ変更用とに切換えて操作できるように構成したり、デュレーション変更用とベロシティ変更用とに役割が分担された2つの操作子を備えてもよい。

【0127】また、上記各実施形態では、リズムパターンテーブルの各ステップに、ステップタイム、デュレーション、ベロシティ係数の三者がセットになって記録されているが、ステップタイムのみ、デュレーションのみ、あるいはベロシティのみのいずれかのリズムパターンテーブルを持ち、他の二者は固定値を用いるようにしてもよい。リズムパターンテーブルとして、デュレーションのみのテーブルを持った場合であっても、時間的に等分された単純なアルペジオ演奏だけでなくさまざまなリズムのアルペジオ演奏が可能となり、ステップタイムのみのテーブルを持った場合であっても、スタッカートやテヌートの効果を、アルペジオ演奏の各ステップ毎に独立して付加できるので、リズムの多彩なニュアンスやアーティキュレーションを付けることができ、ベロシティ係数のみのテーブルを持った場合であっても、アルペジオ演奏の各ステップ毎に強弱を付けることができさまざまなアクセントのアルペジオ演奏が実現するなど、それぞれ音楽的に意味のあるアルペジオ演奏を実現できる。もちろん、上記三者のうちのいずれか二者のリズムパターンテーブルを持ってもよい。

【0128】また、リズムパターンテーブルに関し、上記二者ないし三者のリズムパターンテーブルを持つにあたり、それら二者ないし三者のリズムパターンテーブルをそれぞれ独立して持ち（例えばステップタイムのみのテーブルとデュレーションのみのテーブルとベロシティのみのテーブルとに分けて持ち）、演奏者の好みや、演奏スタイルに応じてそれらを組合せて用いるようにしてもよい。

【0129】なお、リズムパターンテーブルに記憶するデータの種類としてステップタイム、デュレーション、ベロシティ係数以外の種類のデータを記憶するようにしてもよい。例えば、各ステップ毎に音色指定データを記憶すれば各ステップ毎に音色を切り換えることができ、より一層高度なアルペジオ演奏が可能となる。この場合、この音色が、本発明にいう楽音の特質として観念される。さらに、上記各実施形態においては、リズムパターンテーブル、およびスキャンモードテーブルはROM14（図1参照）に固定的に記憶されている旨説明したが、ROM14に代わり、あるいはROM14とともに、RAM13にリズムパターンテーブルないしスキャンモードテーブルを置いてよい。その場合、外部から新たなリズムパターンテーブルないしスキャンモードテーブルをロードして、あるいは、パネル15にリズムパターンテーブルないしスキャンモードテーブルを設定できるような操作子を配置しておいて演奏者がその操作子を操作することにより新たなリズムパターンテーブルな

いしスキャンモードテーブルを定義して、自分の好みに一層適合したアルペジオ演奏を行うことができる。

【0130】

【発明の効果】以上説明したように、本発明のアルペジエータによれば、より音楽的なアルペジオ演奏を簡単な操作で行うことができる。

【図面の簡単な説明】

【図1】本発明のアルペジエータの一実施形態の回路構成図である。

【図2】パネルの構成を示した図である。

【図3】リズムパターンテーブルの基本パターンを示す図である。

【図4】4分音符用のリズムパターンテーブルおよびそのステップ数を示した図である。

【図5】16分音符用のリズムパターンテーブルおよびそのステップ数を示した図である。

【図6】ワルツ用のリズムパターンテーブルおよびそのステップ数を示した図である。

【図7】シャッフル用のリズムパターンテーブルおよびそのステップ数を示した図である。

【図8】ディミニッシュ用のリズムパターンテーブルおよびそのステップ数を示した図である。

【図9】レゲエ用のリズムパターンテーブルおよびそのステップ数を示した図である。

【図10】スキャンモードテーブルの基本パターンを示す図である。

【図11】上昇スキャン用のスキャンモードテーブルおよびそのステップ数を示した図である。

【図12】下降スキャン用のスキャンモードテーブルおよびそのステップを示した図である。

【図13】上昇下降スキャン用のスキャンモードテーブルおよびそのステップを示した図である。

【図14】ランダムスキャン用のスキャンモードテーブルおよびそのステップを示した図である。

【図15】和音スキャン用のスキャンモードテーブルおよびそのステップを示した図である。

【図16】押鍵順スキャン用のスキャンモードテーブルおよびそのステップを示した図である。

【図17】ワルツ用のスキャンモードテーブルおよびそのステップを示した図である。

【図18】レゲエ用のスキャンモードテーブルおよびそのステップを示した図である。

【図19】三味線用のスキャンモードテーブルおよびそのステップを示した図である。

【図20】リズムパターンパラメータを示す図である。

【図21】スキャンモードパラメータを示す図である。

【図22】グループレートパラメータを示す図である。

【図23】ノートバッファを示す図である。

【図24】プレイバッファを示す図である。

【図25】カレントプレイノートナンババッファを示す

図である。

【図 26】ホールドバッファを示す図である。

【図 27】オーダバッファを示す図である。

【図 28】オーダライトカウンタを示す図である。

【図 29】オーダポジションカウンタを示す図である。

【図 30】リズムパターンテーブルポジションカウンタを示す図である。

【図 31】スキャンモードテーブルポジションカウンタを示す図である。

【図 32】クロックカウンタを示す図である。

【図 33】ネクストクロックバッファを示す図である。

【図 34】ノートオフリザベーションバッファを示す図である。

【図 35】エンドフラグを示す図である。

【図 36】ベースノートバッファを示す図である。

【図 37】トップノートバッファを示す図である。

【図 38】ゼネラルプログラムのフローチャートである。

【図 39】初期化処理ルーチンのフローチャートである。

【図 40】リセットスキャンルーチンのフローチャートである。

【図 41】クロック処理ルーチンのフローチャートである。

【図 42】スキャンノートオンルーチンのフローチャートである。

【図 43】スキャンアップルーチンのフローチャートである。

【図 44】スキャンオーダールーチンのフローチャートである。

【図 45】スキャンコードルーチンのプログラムである。

【図 46】スキャンベースルーチンのプログラムである。

【図 47】スキャンコードウィズアウトベースルーチンのプログラムである。

【図 48】ノートオフ予約ルーチンのプログラムである。

【図 49】ベロシティ生成ルーチンのフローチャートである。

【図 50】図 49 に示すベロシティ生成ルーチンで生成されるベロシティの説明図である。

【図 51】ネクストクロック更新ルーチンのフローチャートである。

【図 52】スキャナ更新ルーチンのフローチャートの前

半部分である。

【図 53】スキャナ更新ルーチンのフローチャートの後半部分である。

【図 54】エディットルーチンのフローチャートである。

【図 55】ノートオン処理ルーチンのフローチャートである。

【図 56】ノートオフ処理ルーチンのフローチャートである。

【図 57】デリートオーダールーチンのフローチャートである。

【図 58】バックオーダールーチンのフローチャートである。

【図 59】ホールドオフ処理ルーチンのフローチャートである。

【図 60】リメイクオーダールーチンのフローチャートである。

【図 61】図 2 のパネルに付加されるベロシティポリウムを示した図である。

【図 62】図 49 に示すベロシティ生成ルーチンに代えて採用されるベロシティ生成ルーチンのフローチャートである。

【図 63】図 2 のパネルに付加されるスタイルスイッチを示した図である。

【図 64】図 54 に示すエディットルーチンに代えて採用されるエディットルーチンのフローチャートである。

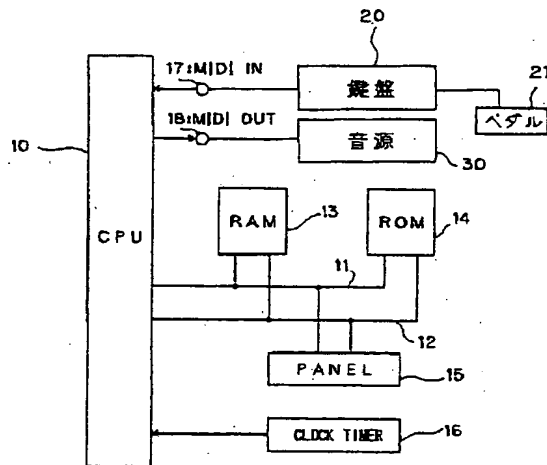
【符号の説明】

- 10 CPU
- 11 アドレスバス
- 12 データバス
- 13 RAM
- 14 ROM
- 15 パネル
- 16 クロックタイマ
- 17 データ入力端子
- 18 データ出力端子
- 20 鍵盤
- 21 ペダル
- 30 音源
- 151 リズムパターン設定用操作子
- 152 スキャンモード設定用操作子
- 153 グループプレート設定用操作子
- 154 ベロシティポリウム
- 155 スタイルスイッチ

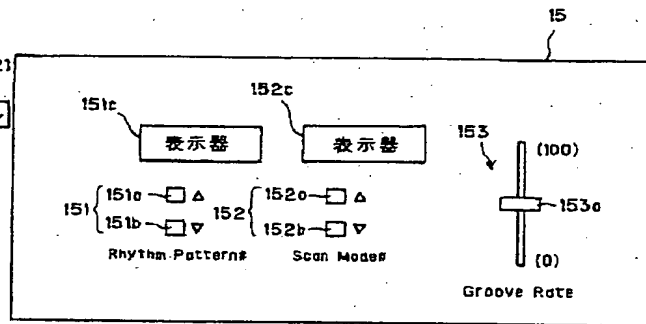
【図 20】

[Rhythm Pattern Parameter]
PRM_RHYTHM ptn. #

【図1】



【図2】



【図3】

[Rhythm Pattern Table]

| RHY_XXX_TBL[n] | Step Time | Duration | Velo. Coef. |
|----------------|-----------|----------|-------------|
| | Step Time | Duration | Velo. Coef. |
| | Step Time | Duration | Velo. Coef. |
| | Step Time | Duration | Velo. Coef. |
| | Step Time | Duration | Velo. Coef. |
| | Step Time | Duration | Velo. Coef. |

RHY_XXX_SIZE: Step Size

【図4】

RHY_4_TBL

| | | |
|-----|-----|-------|
| 9 6 | 9 2 | 1 2 7 |
|-----|-----|-------|

RHY_4_SIZE: 1

【図5】

RHY_16_TBL

| | | |
|-----|-----|-------|
| 2 4 | 2 2 | 1 2 7 |
| 2 4 | 2 2 | 1 0 0 |
| 2 4 | 2 2 | 1 0 0 |
| 2 4 | 2 2 | 1 0 0 |

RHY_16_SIZE: 4

【図6】

RHY_WALTZ_TBL

| | | |
|-----|-----|-------|
| 9 6 | 2 0 | 1 2 7 |
| 9 6 | 2 0 | 1 0 0 |
| 9 6 | 2 0 | 1 0 0 |

RHY_WALTZ_SIZE: 3

【図10】

[Scan Mode Table]

| | |
|------------------|-----------------|
| SMODE_XXX_TBL[n] | Scan Function # |
| | Scan Function # |
| | Scan Function # |
| | Scan Function # |
| | Scan Function # |
| SMODE_XXX_SIZE | Scan Table Size |

【図7】

RHY_SHUFFLE_TBL

| | | |
|-----|-----|-------|
| 3 2 | 3 0 | 1 2 7 |
| 1 6 | 1 4 | 8 0 |
| 3 2 | 3 0 | 1 0 0 |
| 1 6 | 1 4 | 8 0 |

RHY_SHUFFLE_SIZE: 4

【図8】

RHY_DIV_TBL

| | | |
|-----|-----|-------|
| 2 4 | 2 2 | 1 2 7 |
| 2 4 | 2 2 | 1 1 0 |
| 2 4 | 2 2 | 9 5 |
| 2 4 | 2 2 | 8 0 |
| 2 4 | 2 2 | 6 0 |
| 2 4 | 2 2 | 4 0 |
| 2 4 | 2 2 | 3 0 |
| 2 4 | 2 2 | 2 0 |

RHY_DIV_SIZE: 8

【図9】

RHY_REGGAE_TBL

| | | |
|-----|-----|-------|
| 9 4 | 1 | 8 5 |
| 6 8 | 8 | 1 2 7 |
| 3 0 | 4 | 9 6 |
| 9 4 | 1 0 | 8 0 |
| 9 8 | 6 | 1 1 7 |

RHY_REGGAE_SIZE: 5

【図18】

SMODE_REGGAE_TBL

| | |
|-------------|-------------|
| SCAN_B | SCAN_C_WD_B |
| SCAN_C_WD_B | SCAN_C_WD_B |
| SCAN_C_WD_B | SCAN_B |
| SCAN_C_WD_B | SCAN_C_WD_B |

SMODE_REGGAE_SIZE: 5

【図15】

SMODE_CHORD_TBL

| |
|--------|
| SCAN_C |
|--------|

SMODE_CHORD_SIZE: 1

【図17】

SMODE_WALTZ_TBL

| | |
|-------------|-------------|
| SCAN_B | SCAN_C_WD_B |
| SCAN_C_WD_B | SCAN_C_WD_B |
| SCAN_C_WD_B | SCAN_B |

SMODE_WALTZ_SIZE: 3

【図11】

SMODE_UP_TBL

| |
|--------|
| SCAN_U |
|--------|

SMODE_UP_SIZE: 1

【図12】

SMODE_DOWN_TBL

| |
|--------|
| SCAN_D |
|--------|

SMODE_DOWN_SIZE: 1

【図13】

SMODE_UP_DN_TBL

| |
|---------|
| SCAN_UD |
|---------|

SMODE_UP_DN_SIZE: 1

【図14】

SMODE_RANDOM_TBL

| |
|--------|
| SCAN_R |
|--------|

SMODE_RANDOM_SIZE: 1

【図16】

SMODE_ORDER_TBL

| |
|--------|
| SCAN_O |
|--------|

SMODE_ORDER_SIZE: 1

【図21】

[Scan Mode Parameter]

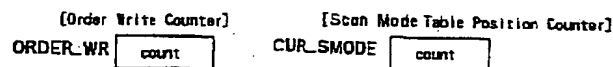
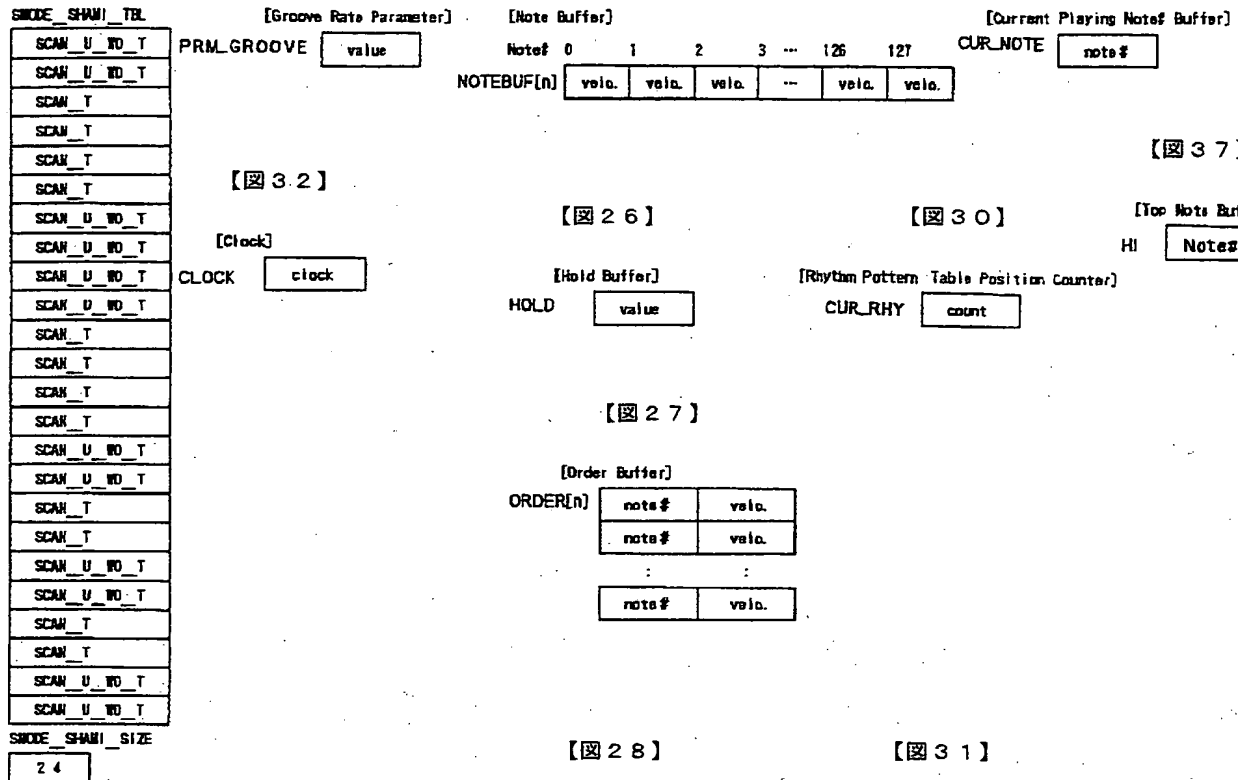
| | |
|------------|--------|
| PRML_SMODE | mode # |
|------------|--------|

【図19】

【図22】

【図23】

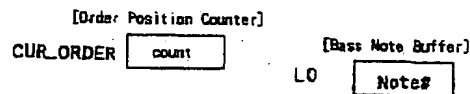
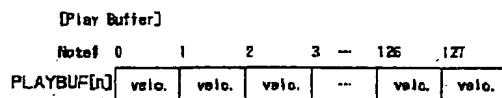
【図25】



【図29】

【図36】

【図24】

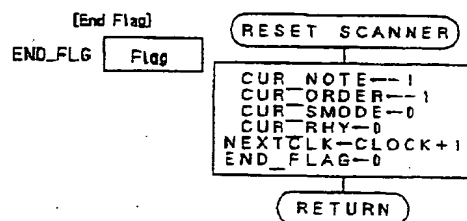
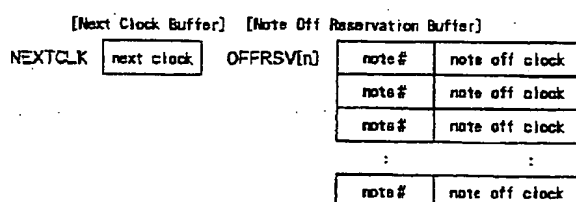


【図35】

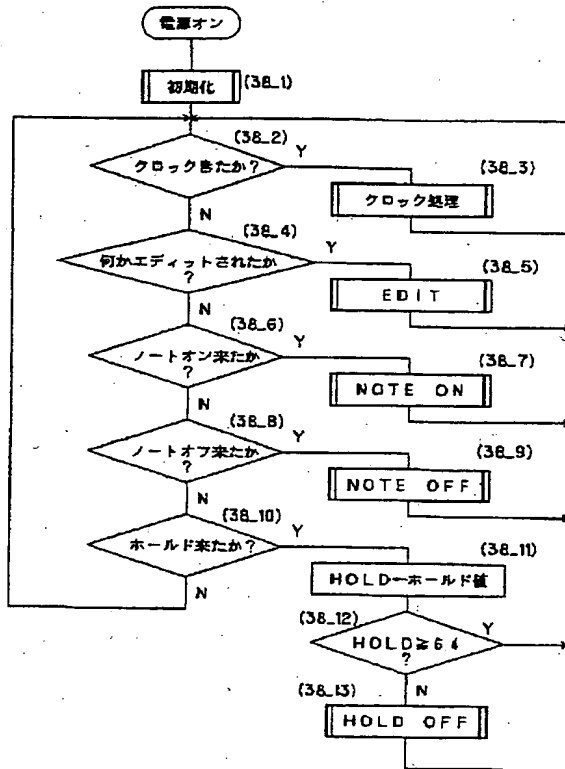
【図40】

【図33】

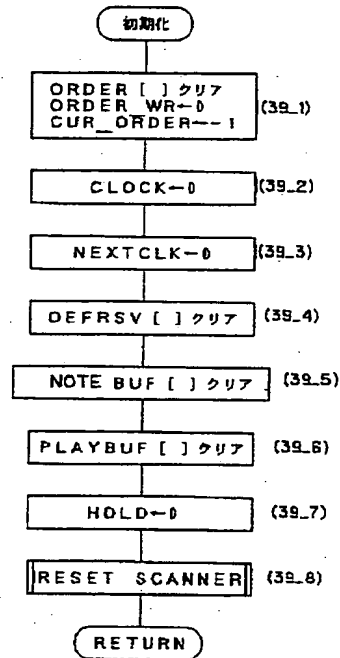
【図34】



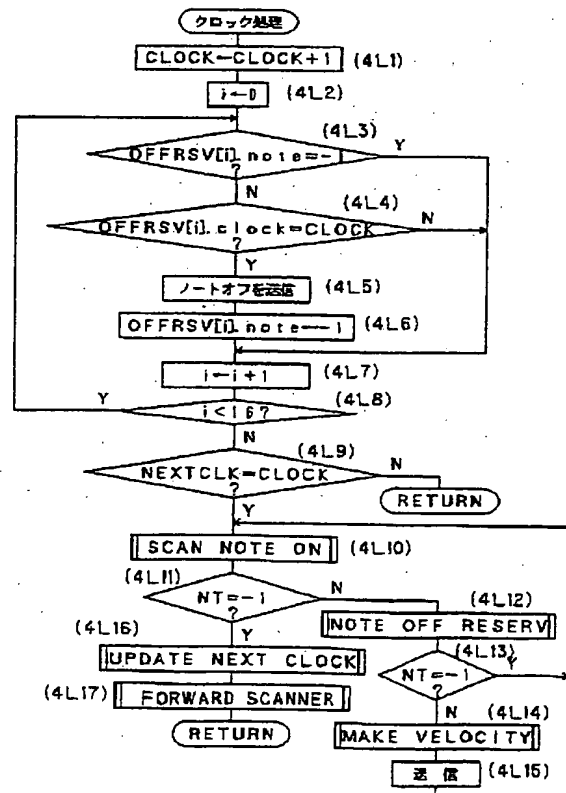
【図 38】



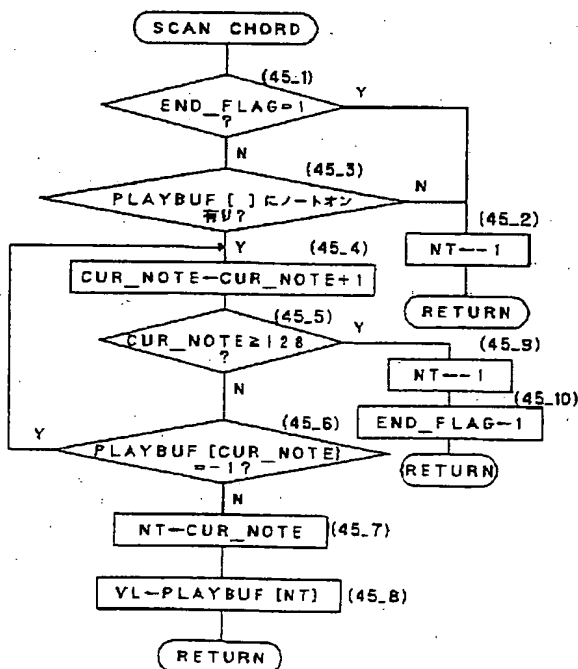
【図 39】



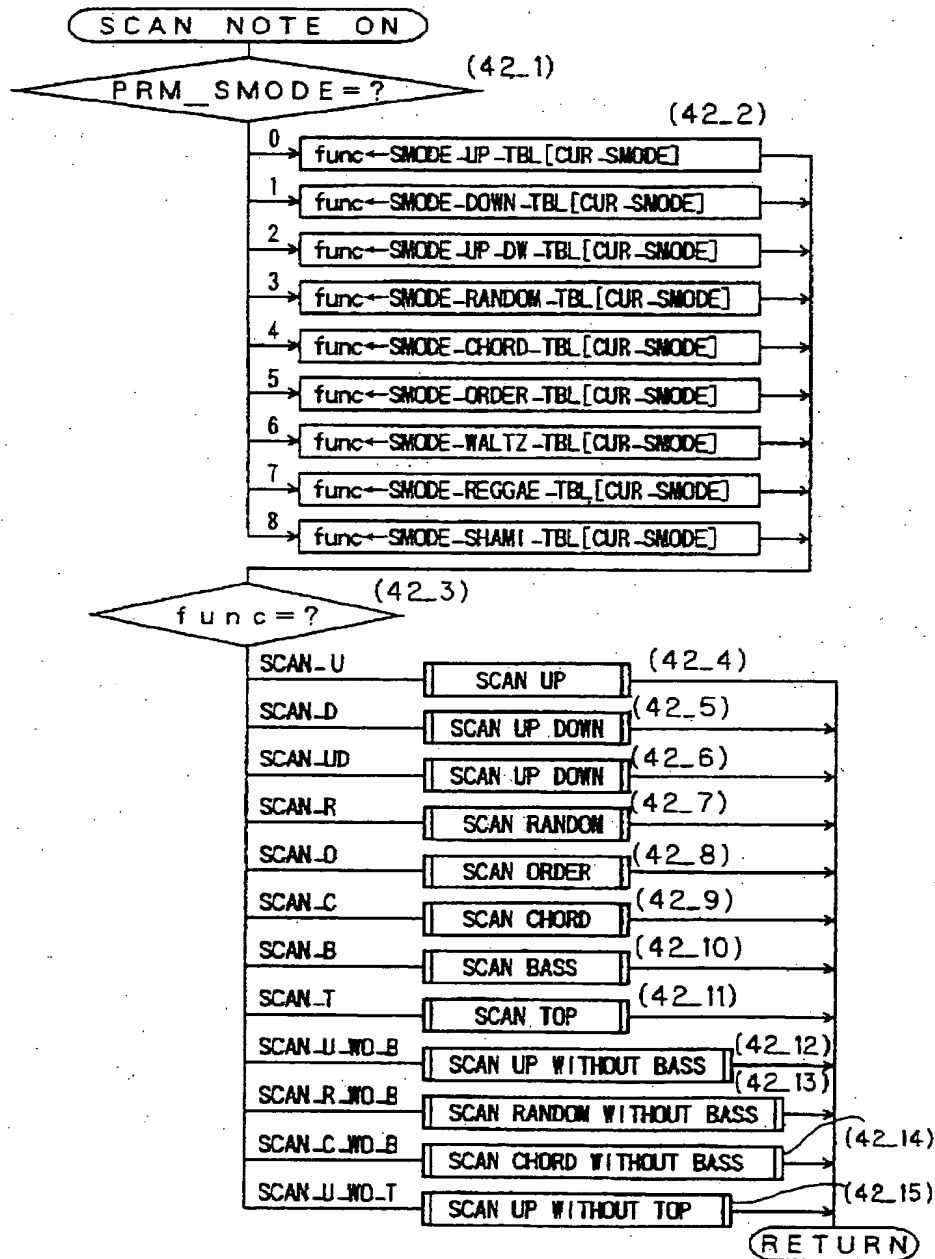
【図 41】



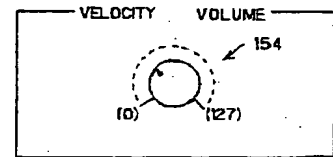
【図 45】



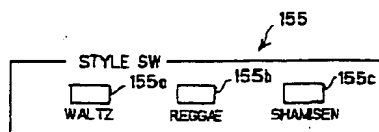
【図 4 2】



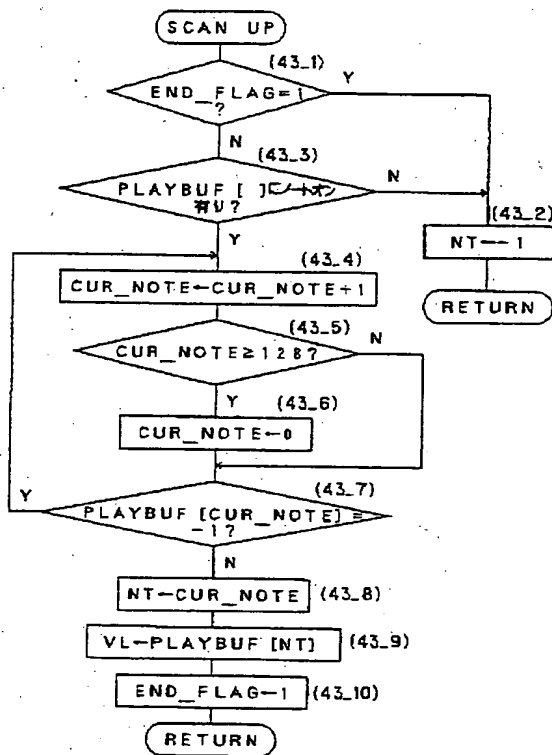
【図 6 1】



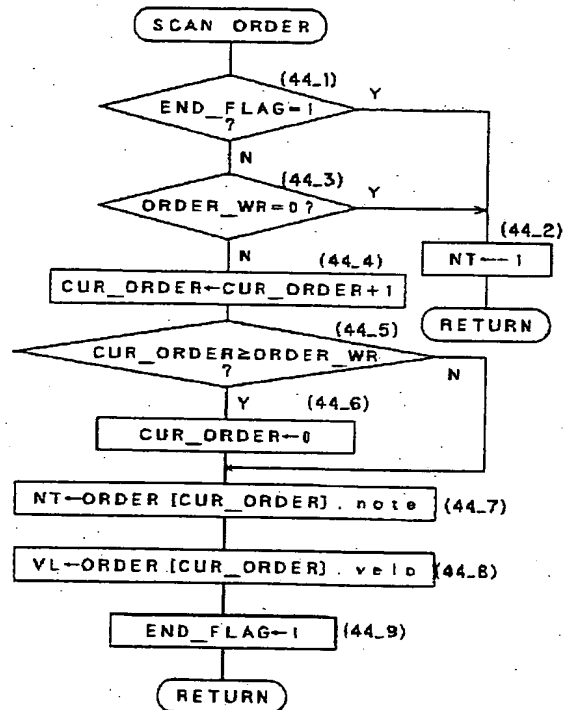
【図 6 3】



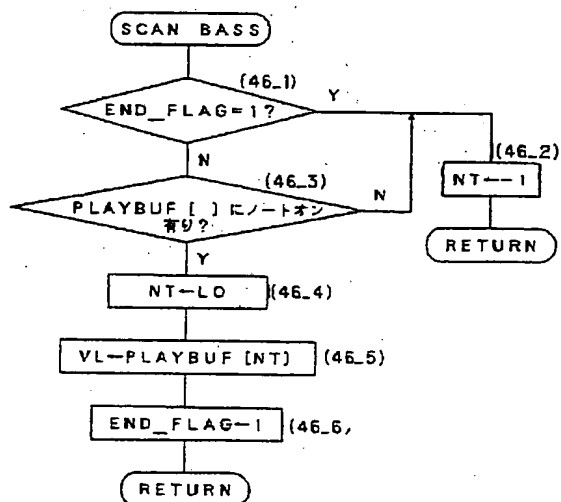
【図43】



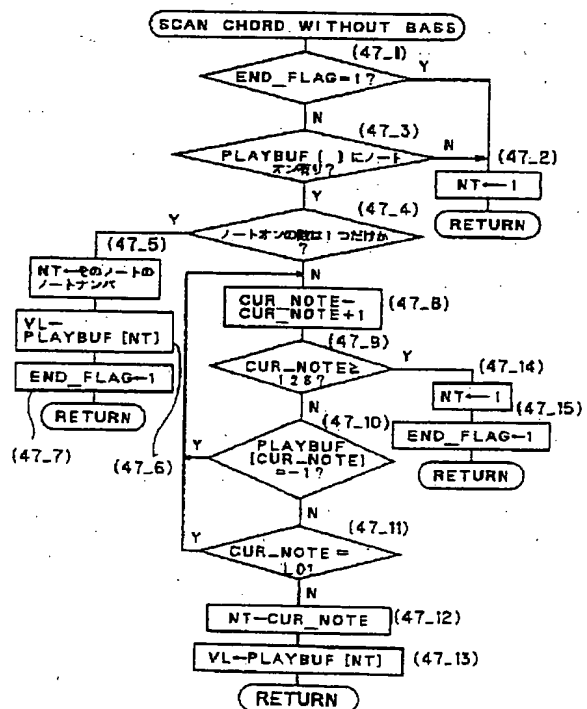
【図44】



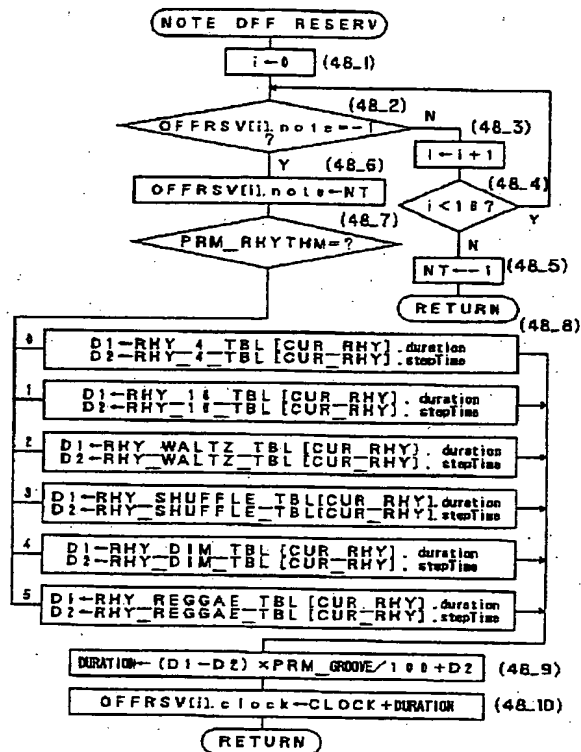
【図46】



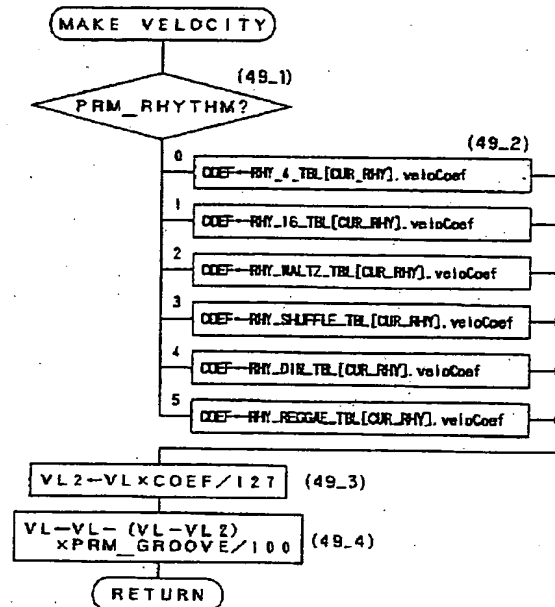
【図47】



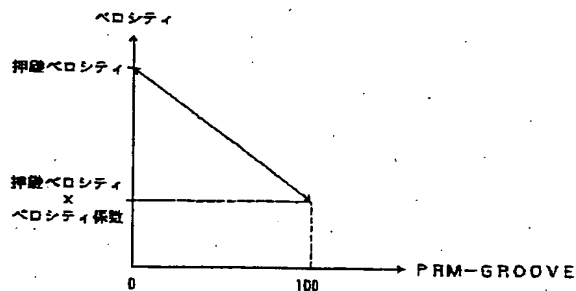
【図48】



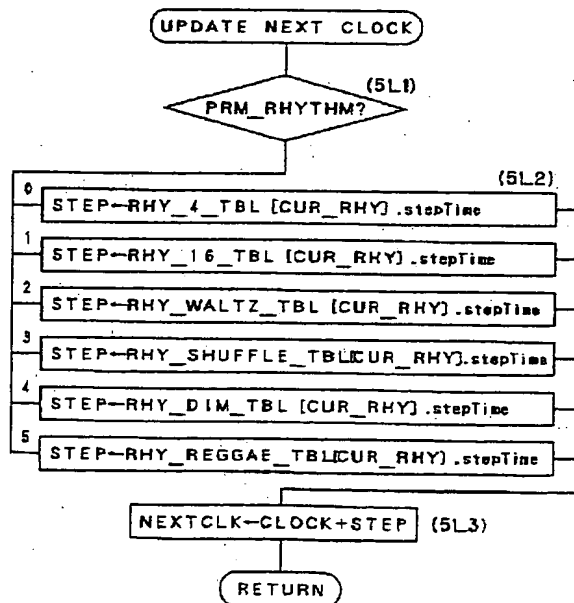
【図49】



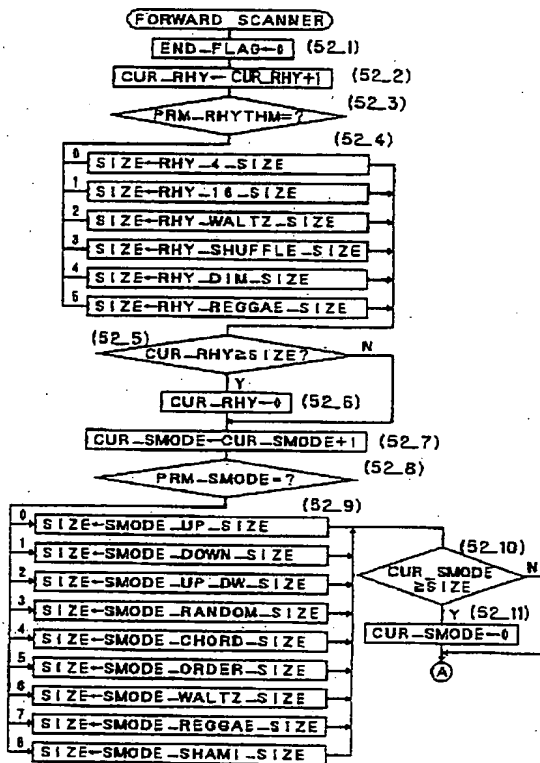
【図50】



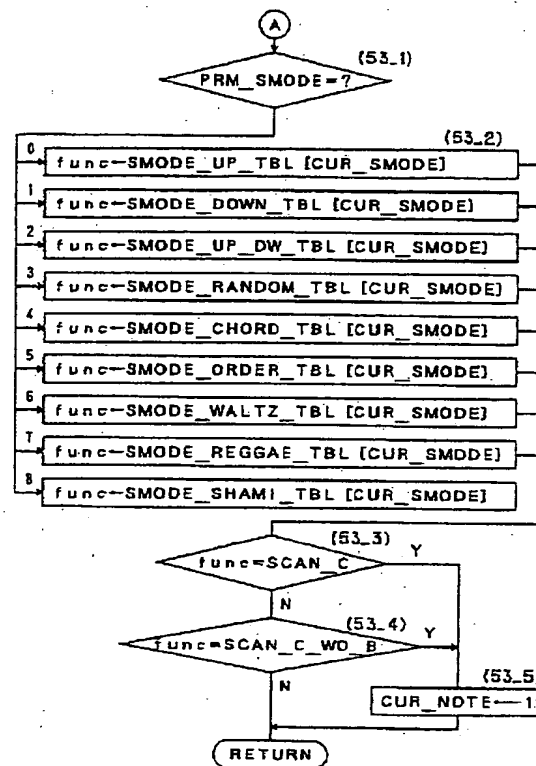
【図51】



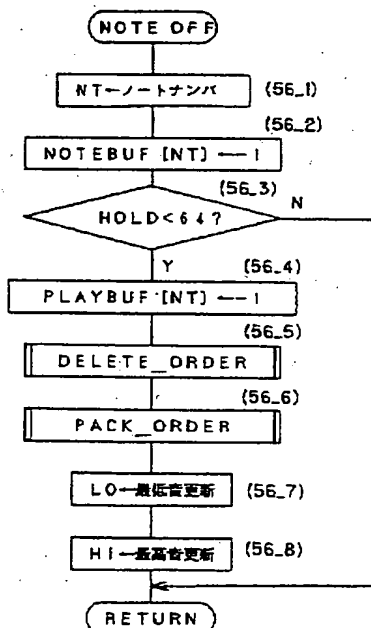
【図 5 2】



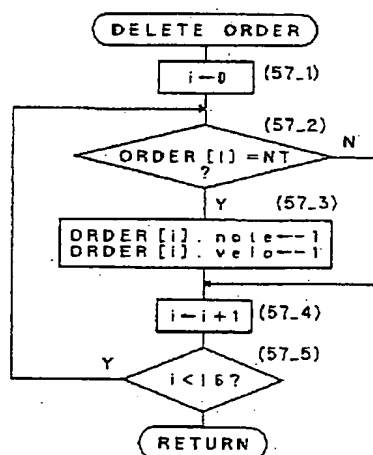
【図 5 3】



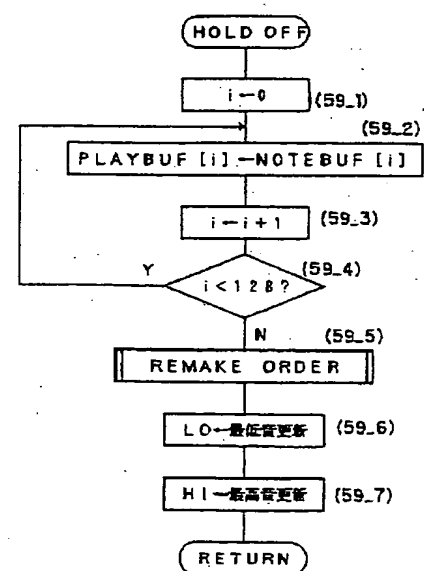
【図 5 6】



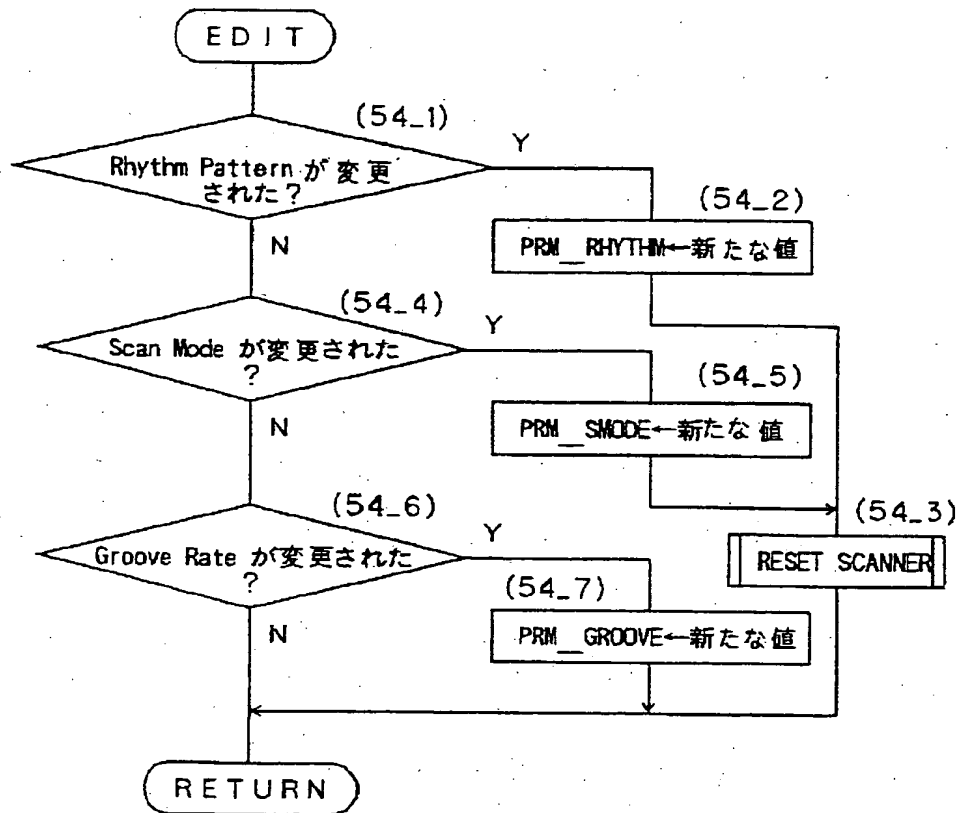
【図 5 7】



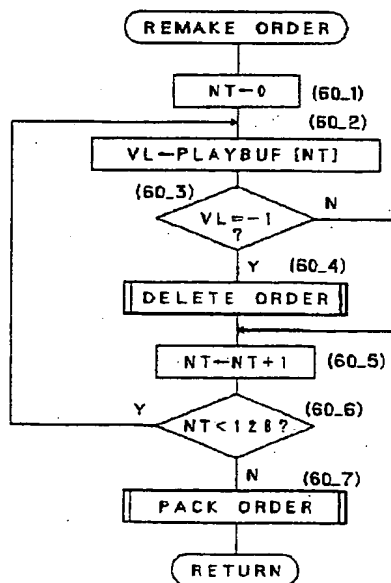
【図 5 9】



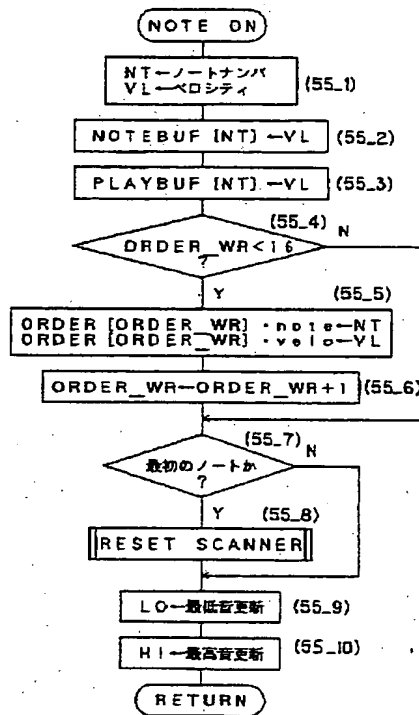
【図 54】



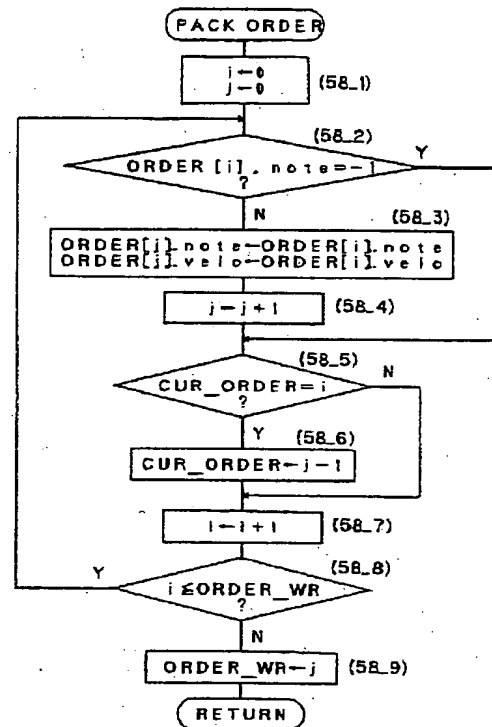
【図 60】



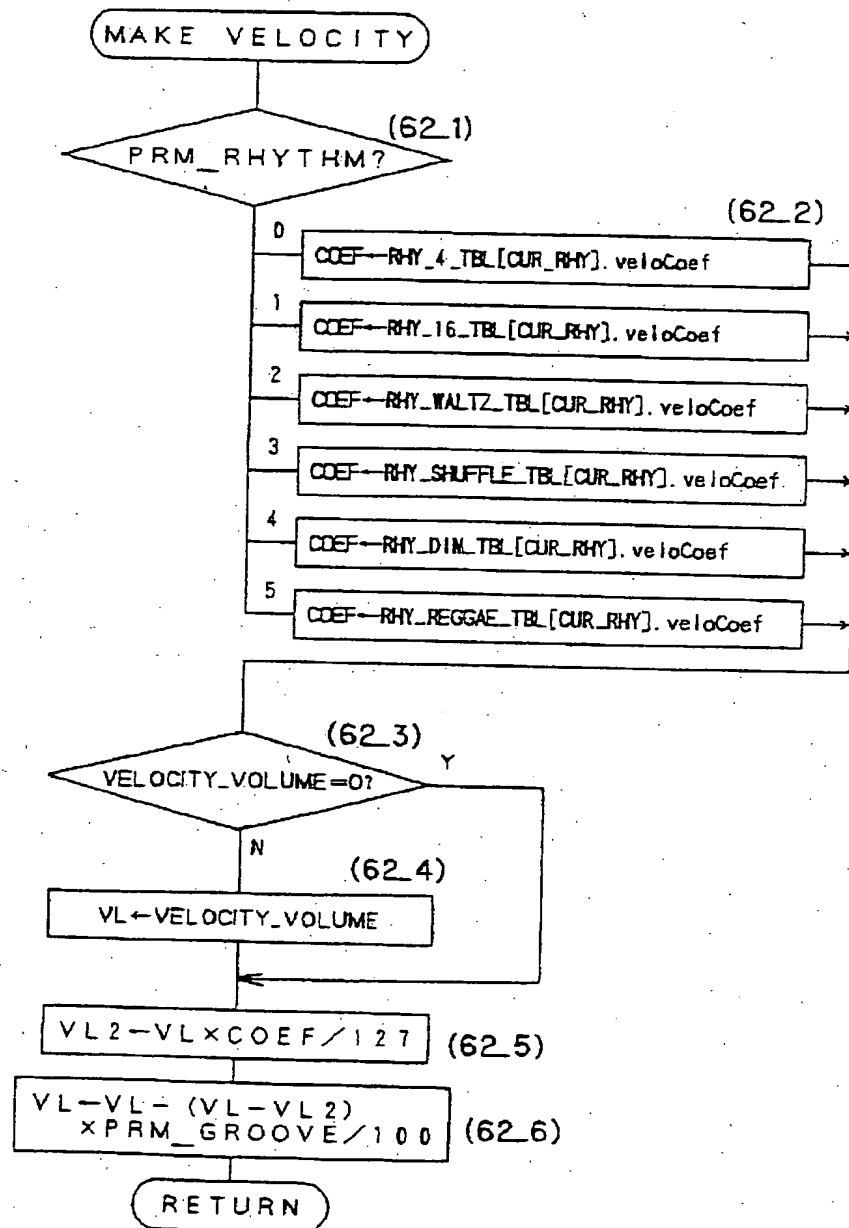
【図55】



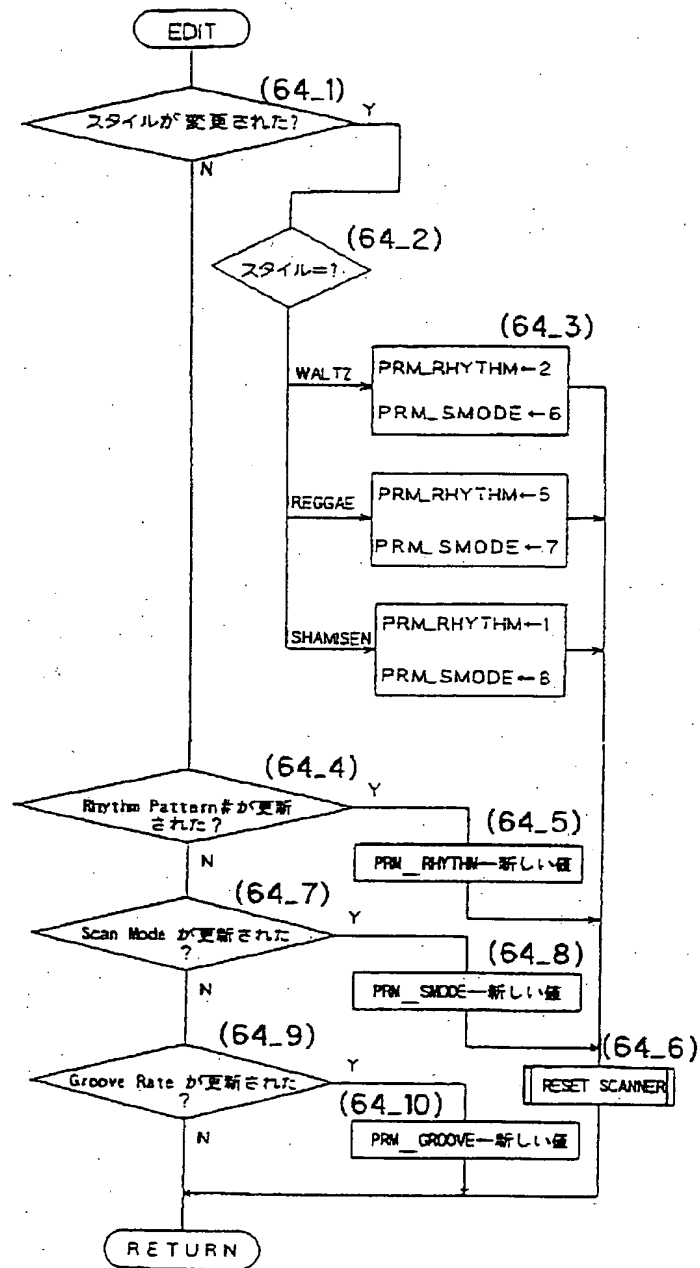
【図58】



【図62】



【図 64】



【公報種別】特許法第17条の2の規定による補正の掲載
【部門区分】第6部門第2区分
【発行日】平成13年12月14日(2001.12.14)

【公開番号】特開平9-90953
【公開日】平成9年4月4日(1997.4.4)
【年通号数】公開特許公報9-910
【出願番号】特願平7-241056
【国際特許分類第7版】

G10H 1/36
1/28

【FI】
G10H 1/36
1/28

【手続補正書】

【提出日】平成13年5月23日(2001.5.23)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正内容】

【特許請求の範囲】

【請求項1】 複数の鍵それぞれに対応する記憶領域を有し、鍵が押鍵されたことを表わす押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込む押鍵情報記憶手段と、

リズムの1ステップ毎に、あるステップと、そのステップに続く次のステップとの間の時間間隔が、1つのステップもしくは複数のステップそれぞれに記録されて成るリズムパターンテーブルを記憶するリズムパターンテーブル記憶手段と、

前記リズムパターンテーブルの各ステップを順次参照するとともに、該各ステップに対応して前記記憶領域を走査し、該走査により検出された押鍵情報に基づく楽音を表わす演奏情報を、前記リズムパターンテーブルの、該走査に対応するステップに記録された時間間隔に従ったタイミングで生成する演奏情報生成手段とを備えたことを特徴とするアルペジエータ。

【請求項2】 複数の鍵それぞれに対応する記憶領域を有し、鍵が押鍵されたことを表わす押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込む押鍵情報記憶手段と、

リズムの1ステップ毎に、そのステップにおける楽音の特質を規定する情報が、1つのステップもしくは複数のステップそれぞれに記録されて成るリズムパターンテーブルを記憶するリズムパターンテーブル記憶手段と、

前記リズムパターンテーブルの各ステップを順次参照するとともに、該各ステップに対応して前記記憶領域を走

査し、該走査により検出された押鍵情報、および前記リズムパターンテーブルの、該走査に対応するステップに記録された、楽音の特質を規定する情報との双方に基づく楽音を表わす演奏情報を生成する演奏情報生成手段とを備えたことを特徴とするアルペジエータ。

【請求項3】 前記楽音の特質を規定する情報が、楽音の発音持続時間を規定する情報であることを特徴とする請求項2記載のアルペジエータ。

【請求項4】 前記楽音の特質を規定する情報が、楽音の発音の強さを規定する情報であることを特徴とする請求項2記載のアルペジエータ。

【請求項5】 前記押鍵情報記憶手段が、鍵が押鍵されたことを表わすとともに押鍵の強さの情報を含む押鍵情報を、押鍵された鍵に対応する記憶領域に、消去自在に書き込むものであって、

前記演奏情報生成手段が、前記走査により検出された押鍵情報に含まれた押鍵の強さの情報に対応する発音の強さの情報を含む演奏情報を生成するものであることを特徴とする請求項1又は2記載のアルペジエータ。

【請求項6】 前記リズムパターンテーブル記憶手段が、複数のリズムパターンテーブルを記憶するものであり、

前記演奏情報生成手段が、前記複数のリズムパターンテーブルの中から選択された1つのリズムパターンテーブルを参照するものであることを特徴とする請求項1又は2記載のアルペジエータ。

【請求項7】 前記リズムパターンテーブル記憶手段が、前記複数のリズムパターンテーブルのうちの少なくとも一部のリズムパターンテーブルを書換え自在に、もしくは新たなリズムパターンテーブルを追加自在に記憶するものであることを特徴とする請求項6記載のアルペジエータ。

【請求項8】 前記演奏情報生成手段が、楽音の特質の変更深さを規定する情報に基づいて変更された楽音を表

わす演奏情報を生成するものであることを特徴とする請求項 1 又は 2 記載のアルペジエータ。

【請求項 9】 前記楽音の特質の変更深さを規定する情報であって、該変更深さが操作の度合に応じて定められてなる情報を生成する操作子を備えたことを特徴とする請求項 8 記載のアルペジエータ。

【請求項 10】 前記楽音の特質の変更深さを規定する

情報が、楽音の発音持続時間の変更の度合を規定する情報であることを特徴とする請求項 8 記載のアルペジエータ。

【請求項 11】 前記楽音の特質の変更深さを規定する情報が、楽音の発音の強さの変更の度合を規定する情報であることを特徴とする請求項 8 記載のアルペジエータ。